March 1988

$3.95 USA (Canada $4.95)

Desktop Mini-supercomputers

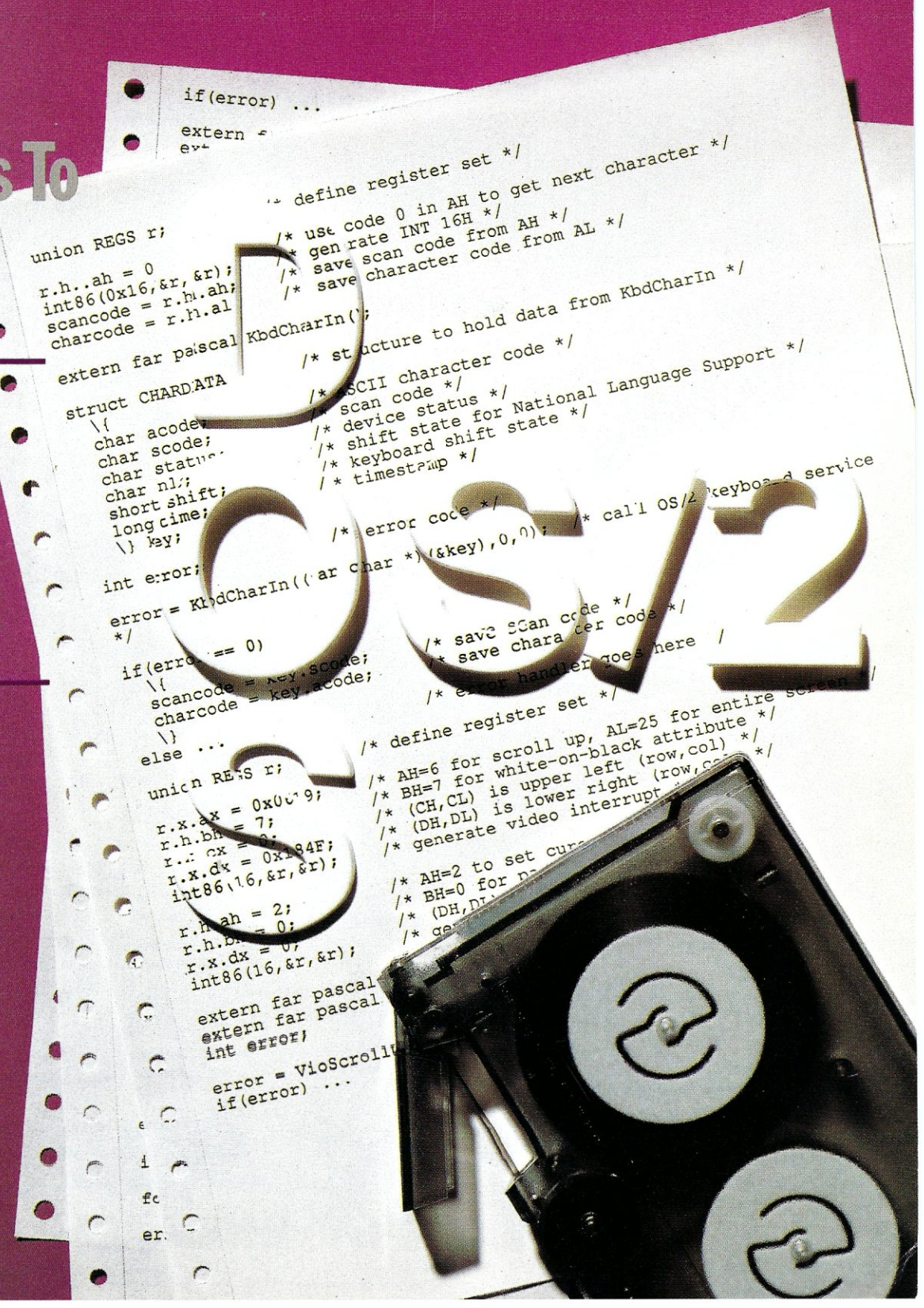
Micro Systems

JOURNAL for the PC Systems Integrator

Porting Programs To OS/2

Novell Print Spooling With Pascal

Real Time OS Reviewed



0 38351 19902 9 03

Powerful Concurrent™ DOS 386

Multiuser Multitasking Operating System

MAXIMUM PERFORMANCE . . . Unleash the Power of the 386!

Concurrent™ DOS uses the power of the 386 to efficiently combine its multiuser, multitasking design with the added value of DOS compatibility. Up to ten users can share the resources of a single system through easy-to-connect serial terminals as configured. Digital Research provides tools for those VARs and OEMs needing to expand beyond ten users.

MAXIMUM COMPATIBILITY . . . Runs Multiuser Applications While Simultaneously Running Many Popular PC-DOS Applications.

The applications running on your current system (Lotus® 1-2-3®, dBase® III, WordPerfect® and many more) are still usable and don't have to be replaced with "work-alikes" and "compatibles." Multiple DOS applications can be run from serial terminals while the system console can execute as many as four applications concurrently.

MINIMUM INVESTMENT . . . Protect Your Development Investment with Easy Migration within the Intel® Microprocessor Family.

Take advantage of developed and proven application software that provides solutions for businesses that range from medical practices to manufacturing floors. A library of Concurrent DOS multiuser applications already exists to meet the diverse requirements of many end-user environments.

FEATURES:

- PC DOS 3.3 Compatible.
- Supports IBM® Personal System/2™ Model 80, Compaq® Deskpro® 386 and 100% Compatibles.
- Executes Multiple Applications on Serial Terminals.
- Simultaneously Executes Up to 255 Different Tasks.
- Supports Multiusers Sharing System Resources.
- Easy to Install and Operate.
- Full Complement of Development Tools and Over 1000 Business Solutions Available.
- AT Performance at the Serial Terminals.
- Serial Port Configurability Up to 38.4K Baud.

Call Today and Receive a FREE Poster of Single-User and Multiuser Concurrent DOS Applications.

1-800-443-4200.

DIGITAL RESEARCH®

Concurrent is a trademark, and Digital Research and the Digital Research logo are registered trademarks of Digital Research Inc. IBM is a registered trademark and Personal System/2 is a trademark of International Business Machines Corporation. Other products and companies mentioned are trademarks, registered trademarks or trade names of their respective companies. Specifications subject to change without notice. Copyright © 1987 Digital Research Inc. All rights reserved.

Create Magic with Blaise.

Turbo C TOOLS™

Magic is easy with Turbo C TOOLS in your bag of tricks. New Turbo C TOOLS™ from Blaise Computing is a library of compiled C functions that allows you full control over the computer, the video environment, and the file system, and gives you the jump on building programs with Borland's new C compiler. Now you can concentrate on the creative parts of your programs.

The library comes with well-documented source code so that you can study, emulate, or adapt it to your specific needs. Blaise Computing's attention to detail, like the use of function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes Turbo C TOOLS the choice for experienced software developers as well as newcomers to C.

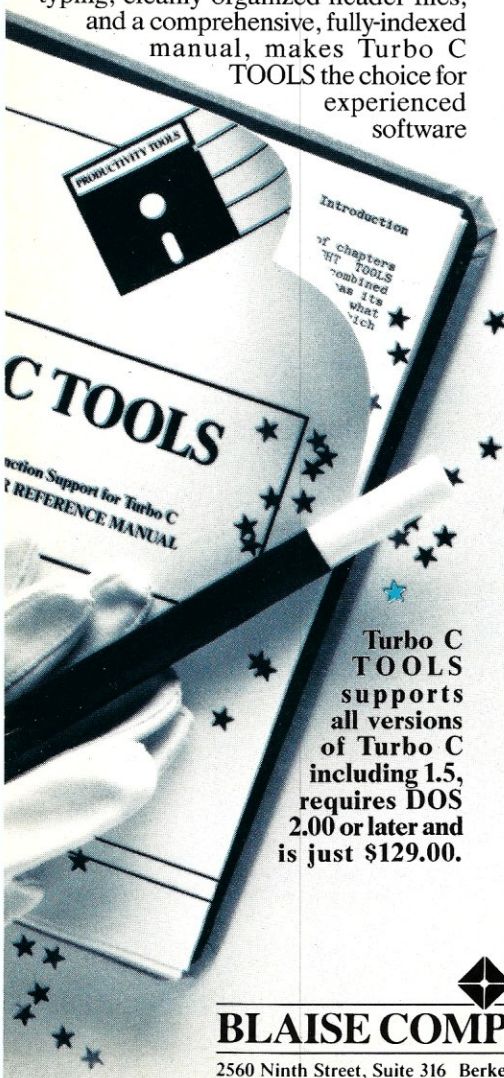
Turbo C TOOLS provides the sophisticated, bullet-proof capabilities needed in today's programming environment, including removable windows, "sidekickable" applications, and general interrupt service routines written in C.

The functions contained in Turbo C TOOLS are carefully crafted to supplement Turbo C, exploiting its strengths without duplicating its library functions. As a result you'll get functions written predominantly in C, that isolate hardware independence, and are small and easy to use.

Turbo C TOOLS embodies the full spectrum of general purpose utility functions that are critical to today's applications. Some of the features in Turbo C TOOLS are:

- **WINDOWS** that are stackable and removable, that have optional borders and a cursor memory, and that can accept user input.
- **INTERRUPT SERVICE ROUTINE** support for truly flexible, robust and polite applications. We show you how to capture DOS critical errors and keystrokes.
- **INTERVENTION CODE** lets you develop memory resident applications that can take full advantage of DOS capabilities. With simple function calls, you can schedule a Turbo C function to execute either when a "hot key" is pressed or at a specified time.
- **RESIDENT SOFTWARE SUPPORT** lets you create, detect, and remove resident utilities that you write with Turbo C TOOLS.
- **FAST DIRECT VIDEO ACCESS** for efficiency, and support for all monitors including EGA 43-line mode.
- **DIRECTORY AND FILE HANDLING** support let you take advantage of the DOS file structure, including volume labels and directory structure.

Blaise Computing Inc. has a full line of support products for Turbo C, Microsoft C, Turbo Pascal and Microsoft Pascal. Call today for details, and make magic!



Turbo C TOOLS supports all versions of Turbo C including 1.5, requires DOS 2.00 or later and is just $129.00.

THE BLAISE M E N U

Turbo POWER SCREEN *$129.00*
NEW! General screen management; paint screens; block mode data entry or field-by-field control with instant screen access. Now for Turbo Pascal 4.0, soon for C and BASIC.

Turbo POWER TOOLS PLUS *$129.00*
NEW VERSION! Now supports Turbo Pascal 4.0. Screen, window, and menu management including EGA support; DOS memory control; ISRs; scheduled intervention code; and much more.

Turbo ASYNCH PLUS *$129.00*
NEW VERSION! Now supports Turbo Pascal 4.0. Interrupt driven support for the COM ports. I/O buffers up to 64K; XON/XOFF; up to 19.2K baud; modem control and XMODEM file transfer.

C TOOLS PLUS *$129.00*
Full spectrum of general service utility functions including: windows; menus; memory resident applications; interrupt service routines; intervention code; and direct video access for fast screen handling. Specifically designed for Microsoft C 5.0 and QuickC.

ASYNCH MANAGER *$175.00*
Full featured interrupt driven support for the COM ports. I/O buffers up to 64K; XON/XOFF; up to 9600 baud; modem control and XMODEM file transfer. For Microsoft C and Turbo C or MS Pascal.

PASCAL TOOLS/TOOLS 2 *$175.00*
Expanded string and screen handling; graphics routines; memory management; general program control; DOS file support and more. For MS-Pascal.

KeyPilot *$49.95*
"Super-batch" program. Create batch files which can invoke programs and provide input to them; run any program unattended; create demonstration programs; analyze keyboard usage.

EXEC *$95.00*
NEW VERSION! Program chaining executive. Chain one program from another in different languages; specify common data areas; less than 2K of overhead.

RUNOFF *$49.95*
Text formatter for all programmers. Written in Turbo Pascal: flexible printer control; user-defined variables; index generation; and a general macro facility.

TO ORDER CALL TOLL FREE
800-333-8087
TELEX NUMBER - 338139

YES! I want to make magic!
Enclosed is $_____ for _____ copies of _____.
☐ Please send me more information on your products.
CA residents add Sales Tax. Domestic orders add $4.00 for UPS shipping, $10.00 for Federal Express standard air.
Name:_____ Phone: (_____) _____-_____
Address:_____ State:_____ Zip:_____
City:_____ Exp. Date:_____
VISA or MC#:_____

BLAISE COMPUTING INC.
2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

Microsoft and QuickC are registered trademarks of Microsoft Corporation. Turbo C and Turbo Pascal are registered trademarks of Borland International.

Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it—the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL,® the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations. In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only $795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL13.

NOVELL

For software solutions, you should be seeing red.

*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.

Micro Systems
JOURNAL for the PC Systems Integrator

March 1988
Vol. 4 . No. 3

FEATURE ARTICLES

Scanning the OS/2 Horizon
A quick look at how OS/2 implements multitasking.
by David E. Cortesi 23

Converting DOS Programs to OS/2 Protected Mode
Guidelines for porting your C and assembly language programs from DOS to OS/2.
by David Schmitt 24

Spooling on Novell Networks with Pascal
Solving printer spooler control problems on Novell local area networks using Turbo Pascal.
by Ed Rought and Tom Hoops 36

PRODUCT REVIEWS

MicroWay's RTOS
An implementation of Intel's Real Time Operating System for the IBM PC/XT and compatibles.
by Stuart Jones 44

COLUMNS

From the Editor's Desk *by Sol Libes*
OS/2 Has Arrived! Or Has It? 6

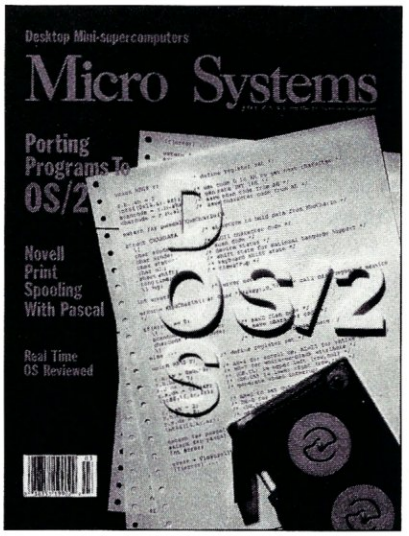
The C Forum *by Don Libes*
Pointers to Functions 10

Turbo Pascal Corner *by Stephen Randy Davis*
Virtual Memory Techniques: Part 1—Arrays of Pointers 18

LANscape *by Mike Cherry*
3270 gateways—Connecting IBM Mainframes to PCs 52

The Scientific Computer User *by A.G.W. Cameron*
Mini-supercomputer on the Desktop 60

The Public Domain Software Forum *by Charles Strom*
A DOS Enhancement and a Powerful Printer Utility 66



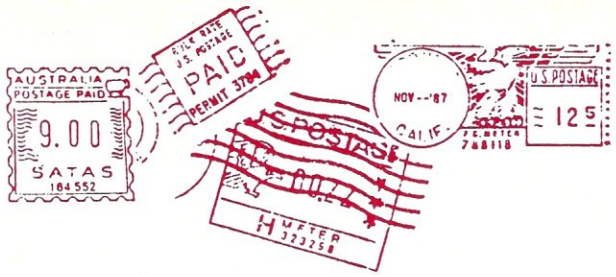
About the cover: The first version of IBM's Operating System 2 has finally arrived, and many programmers are busy porting DOS applications to run under OS/2's protected mode multiprocessing facility. In this issue, we present two articles to assist readers in preparing for the coming of OS/2.
Cover photograph by Michael Carr

DEPARTMENTS

There Is Mail 4
New Products 68
Classifieds 72
Advertiser's Index 72

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points. **POSTMASTER:** send address changes (form 3579) to Micro/Systems Journal, Box 3713, Escondido, CA 92025. **Change of Address:** Please send old label and new address to Micro/Systems Journal, Box 3713, Escondido, CA 92025. **Customer Service:** For subscription problems, call: (800) 321-3333; In California call (619) 485-9623. **Subscription Rates:** U.S., $29.95 for one year, $56.97 for two years. Canada and Mexico add $17 per year for airmail, $7 per year for surface mail. Other countries add $28 per year for airmail.
Entire contents copyright © 1988 by M & T Publishing, Inc., unless otherwise noted on specific articles. All rights reserved.

There is Mail



We welcome your letters, with their comments, compliments, criticism, and suggestions. We do not have the staff to answer all letters personally. All letters become the property of Micro/Systems Journal and may be subject to editing. We do not print letters that do not include a name and address.

Please send your letters to: Micro/Systems Journal, *501 Galveston Drive, Redwood City, CA 95063.*

DB-9 Serial Port Connector Clarified

Dear Editor:

I found your article "Dealing With the DB-9 Serial Port" (*M/SJ*, November/December 1987) interesting and informative, but slightly in error.

When dealing with "S-sub" connectors, the letter immediately following the "D" denotes the connector shell size and the last letter in the description denotes the "sex"—"P" for plug and "S" for socket. Contrary to IBM's own documentation they do not use a "DB-9" but use a "DE-9P" on their systems. I believe it almost impossible for anyone to find a "DB" connector with only 9-pins that will mate with IBM's 9-pin connector.

The standard connector configuration is "DE-9P" for a male connector, and a "DE-9S" for the female connector. "DB-9P" and "DB-9S" are also valid.

Also, according to my schematics, you have pins 2 and 3 swapped as to which is input and which is output. Pin-2 should be the RxD input and Pin-3 the TxD output. Remember, IBM likes to configure its systems as DTE devices.

Bruce D. Fischer, Engineer
Altos Computer Systems
Newark, Calif.

From The Editor:
My apologies. Bruce is correct in that the functions of pins 2 and 3 on the IBM serial interface DB-9 connector are just the reverse of their functions on IBM's serial DB-25 connector. IBM is noted for not being consistent in designating connector pin functions.

Wait States & Refresh Explained

Dear Editor:

I believe that Jeff Duntemann is in error concerning his discussion of the concepts of "wait states" and "refresh" ("Intel's iSBC 386AT," *M/SJ*, November/December 1987). His statement, "Once you read data from a DRAM, you must allow a period of time to pass before reading it again to refresh it at least once between reads," is erroneous.

Wait states are needed with any microprocessor when its clock rate is high enough, so that the time from valid memory address to data in/out is less than the access time of the memory chip. This access time is independent of the type of RAM being used. It is the same for static RAM as it is for Dynamic RAM.

Refresh means that each "row" of a RAM memory array must be addressed within a certain period of time or the memory circuits will lose their data. This is usually 10–100 milliseconds for all row addresses to be accessed. Usually, the circuitry that generates the refresh cycles are buried within the microprocessor or memory control hardware.

I hope that this explanation clarifies the difference between these two parameters.

Arnold S. Berger, PhD
R & D Project Manager
Hewlett-Packard Corp.
Colorado Springs, Colo. □

Micro Systems
JOURNAL for the PC Systems Integrator

EDITORIAL
Founder and Editor Sol Libes
Technical Editors Stephen R. Davis, Don Libes
Associate Editors Lennie Libes, Susan Libes
Contributing Editors A.G.W. Cameron, Michael Cherry, Ian F. Darwin, P.L. Olympia, Charles H. Strom
Consulting Editor Michael Swaine
Managing Editor Thomas M. Woolf

PRODUCTION
Art & Production Director Larry Clay
Art Director Kobi Morgan
Assistant Art Director Barbara Mautz
Typesetter Lorraine Buckland

CIRCULATION
Director of Circulation Maureen Kaminski
Subscription Supv. Kathleen Shay
Newsstand Sales Coordinator Sarah Frisbie
Fulfillment Coordinator Francesca Martin

ADMINISTRATION
Vice President Finance & Operations Kate Wheat
Business Manager Betty Trickett
Accounting Supv. Mayda Lopez-Quintana
Accounts Payable Asst. Luanne Rocklewitz
Accts. Receivable Asst. Michael Eyraud

ADVERTISING
Advertising Director Richard Mixter
National Account Mgr. Dwight Schwab
National Account Mgr. Tami Brenton (415) 886-1957
Advertising Coordinator Shaun Hooper

M&T Publishing, Inc.
Chairman of the Board Otmar Weber
Director C.F. Von Quadt
President and Publisher Laird Foshay
V.P. of Publishing William P. Howard

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

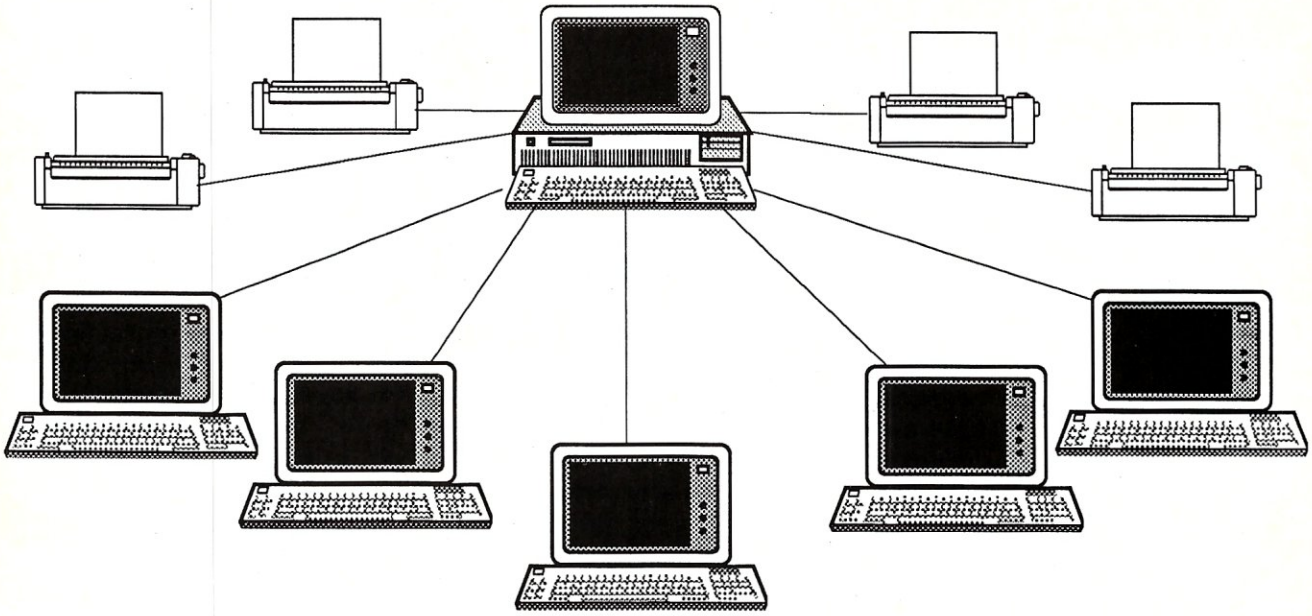
Article Submission: If you have a specific area of expertise or interest and would like to contribute, please write Micro/Systems Journal, P.O. Box 1192, Mountainside, NJ 07092; (201) 522-9347, or contact M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Please do not submit articles without first contacting the editors. Author's guidelines available upon request.

Correspondence: Please send letters to the editor to Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063. Other editorial correspondence may also be directed to P.O. Box 1192, Mountainside, NJ 07092. The editors may also be reached via MCI Mail (SLIBES or MSJ).

Advertising Rates: Available upon request. Call (415) 366-3600 or write to: Advertising Department, Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063.

386-DOS™

A MULTI-USER DOS OPERATING SYSTEM FOR VARS & SYSTEM INTEGRATORS



386-DOS™, a Concurrent™-based operating system from **Concurrent Controls, Inc.** is a reliable, high-performance multiuser DOS solution for **80386**-based microcomputers that combines power and flexibility with **PC-DOS** and **Concurrent** compatibility.

80386 POWER

- Supports up to **16 MB of RAM.**
- Have **1-19 users**, additional printers/devices.
- Run **PC-DOS programs** on dumb terminals.
- **Multiuser system** is **protected** against individual user's program **errors/crashes.**
- Use **intelligent I/O boards** to handle all I/O operations for **increased performance.**
- Supports **large hard disks** up to **512 MB, multiple DOS partitions** up to **512 MB.**

SUPPORT

- **Prompt, professional** phone support.
- **Dealer, system integrator & programmer** support provided by specialists in each area.

FLEXIBILITY

- Use **your choice** of serial port board.
- User configurable IRQ lines, polled or interrupt driven, and port buffer sizes.
- Use **MS-Net** versions of PC-DOS programs.
- Supports **PC-DOS record/file locking.**
- **Add users, connect CPUs** at any time.

EASE OF USE

- Familiar **PC-DOS** interface and commands.
- Menu driven installation and maintenance.
- Single screen **CONFIG** sets all serial port parameters and checks for **accuracy.**
- Use cost-efficient dumb terminals for users - **avoid cost & complication** of **networks.**

386-DOS™ allows you to provide a multiuser DOS solution at a fraction of the costs and complications of a PC network. 386-DOS™ has been tested with a wide range of applications based on Clipper, Foxbase, DBase III, and is also compatible with older Concurrent applications. With 386-DOS your installations can grow from a multitasking, single-user station to a large multiuser installation on the same operating system — reducing learning curves, training time, and total system costs.

Concurrent is a trademark of Digital Research Inc. 386-DOS is a trademark of Concurrent Controls, Inc., other names used here may be trademarks or registered trademarks of other companies.

CONCURRENT CONTROLS, INC.(415)648-2174

3770 24th Street, Suite 206, San Francisco, CA 94114 FAX (415)648-0340

Dealer, distributor and OEM inquiries welcomed

From The Editor's Desk

by Sol Libes

OS/2 Has Arrived! Or Has It?

OS/2 is here, but the question is who is buying it? IBM began shipping OS/2 Version 1.0 in early December, and Zenith started shipping soon after. Compaq and Wyse Technologies will also make implementations for their 286 and 386 based systems available in the immediate future. However, most other system manufacturers are currently adopting a wait-and-see posture. At some point, however, all computer companies will be offering OS/2 as an option while bundling DOS with most systems.

There is no software currently available that takes advantage of OS/2-enhanced features, such as multitasking. Most current applications are merely ports of software already running under DOS. And there is a strong likelihood that we will not see any really worthwhile OS/2 applications until OS/2 Version 1.1 containing the Presentation Manager is released in the fall.

So who is buying OS/2? Mostly software developers and users taking advantage of IBM's $200 upgrade for owners of PC-DOS 3.x (the regular price is $325). It is estimated that there are more than 3,000 programmers actively working on OS/2 application software. This large number of software developers is bound to produce some worthwhile OS/2 application products, and these products should begin to appear very late this year, with market momentum building in 1989 and 1990.

Sales of 286-based systems have already exceeded those of 8088/8086-based systems, and sales of the latter are expected to begin tapering off gradually. Some time in the early 1990s, sales of 386-based systems will also probably exceed those of 8088/8086-based systems, and the 386 is really the performance platform needed for OS/2.

The question is, which 286/386 operating system will dominate? Will it be OS/2, a new version of DOS with Windows, or UNIX with Xwindows? And what about all those operating systems, like PC/MOS-386, Concurrent DOS, Theos, and Pick?

One thing is for sure, we are in for interesting times and *Micro/Systems Journal* will be here to guide you through the changes. □

Sol Libes

News & Views

Random Rumors & Gossip

IBM and **Microsoft** are expected to release one or more new versions of DOS this year. The new DOS generation(s) will add several new features, among them a new user interface, hard disk storage greater than 32 MB, limited multitasking via special software drivers, and built-in support for memory greater than 640K via the Lotus/Intel/Microsoft Expanded Memory Specification (LIM EMS Version 4.0). Some of these features have already been implemented on versions of DOS offered by Compaq, Zenith, and other system manufacturers.

IBM's first PS/2 plug-in card utilizing the multimaster feature of the Micro Channel Architecture (MCA) is expected to become available shortly. Rumors are that it will be an RT RISC coprocessor card for the Model 80. It takes over the MCA, relegating the 386 to the role of an I/O processor. IBM is expected to follow this with a 360 multimaster-type coprocessor card. The question is whether IBM will sell these as add-in cards or as complete systems.

Dataquest, a respected market researcher, estimates that by the end of 1987 there were 12 million PC clones installed worldwide. Add that to about 8 million IBM PCs and you get about 20 million machines in operation. This figure is expected to double by the end of next year.

AT&T and **Sun Microsystems** have disclosed that they are jointly developing a new version of UNIX that will have real-time capabilities, concurrency, fault tolerance, parallel processes, and contain various security features.

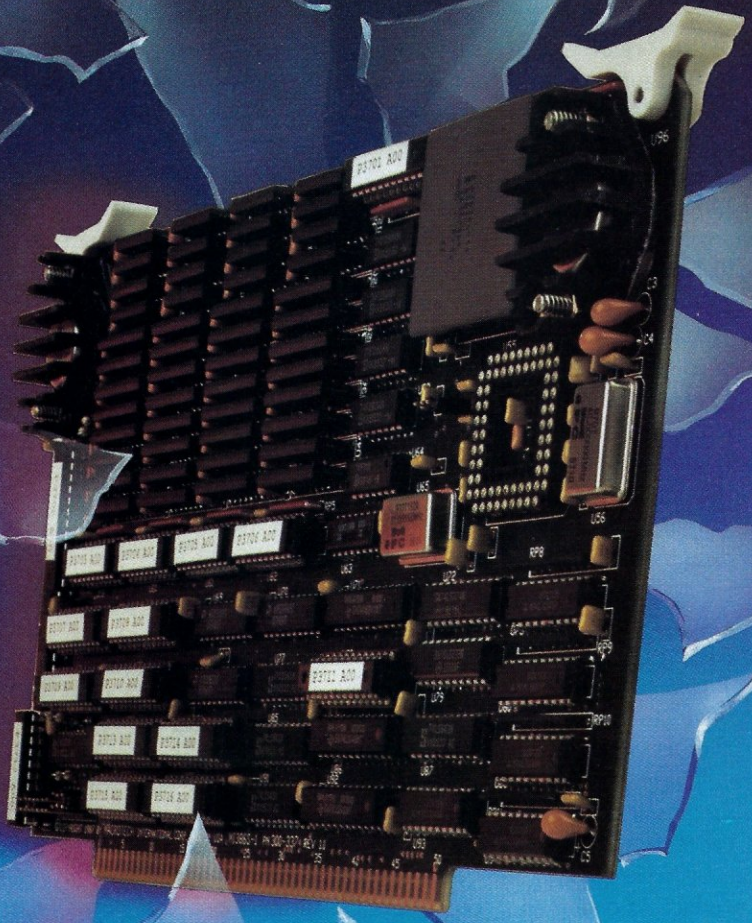
Prices of modems using the CCITT V.32 9,600 kbps standard are coming down as more manufacturers enter this market. **Universal Data Systems** now sells a basic V.32 modem for a list price of $1,595. Multispeed operation and other features increase the cost. Non-compatible 9,600-baud modems from vendors such as **Hayes** and **U.S. Robotics** are in the $1,000 to $1,200 range.

Intel reports that demand for its 80386 microprocessor has grown at a much faster rate than company executives had originally projected, and that it has already been designed into 400 products. Intel now has four 386 production lines and is still backordered. However, there are whispers emanating from several IC houses that they will soon introduce RISC processor 80386 clone chips.

Microsoft is rumored readying a Macintosh coprocessor card that supports OS/2.

Sharp is the first company to go into actual production on an erasable optical drive, the JY-500. Sharp's JY-

SHATTER THE PERFORMANCE BARRIER



PLUG 386 POWER INTO YOUR S100

Announcing the MI386S, the 80386 satellite board for your S100. Drop one into your system and watch it take off. Or add several and prepare for a performance explosion.

It's packed with a full megabyte of 32 bit wide, dual ported, 100 ns, 4-way interleaved dynamic RAM, a 16 MHz 80386 processor, and an optional 80387 math coprocessor.

The powerful MI386S software, compatible with Concurrent DOS, provides a comprehensive and well-honed interface to your system.

Run more programs and more users faster than ever before. A must for the multi-user system. A boon to the single user.

The MI386S. Another first from Macrotech. For more information contact Macrotech International Corporation, 21018 Osborne, Bldg. 5, Canoga Park, CA 91304.

FAX 818-700-1982 • TELEX 910-997-0653

818-700-1501 • 800-824-3181

MACROTECH

Concurrent DOS is a trademark of Digital Research Inc.

500 uses a 5¼-inch removable disk cartridge storing 422 MB. The drive costs $12,000 and the disks are $675 each. It is rated for 1 million erasure/rewrite cycles over 10 years.

Apple Rumors

Apple is expected to release its UNIX operating system for the Mac II very soon, about six months later than promised. Called "A/UX," it is expected to sell for about $1,500 and be a System V implementation with 4.2 BSD extensions.

Apple has also disclosed that it has begun work on a new Mac operating system to be released in about three-to-four years. Aimed at competing with IBM's OS/2 Extended Edition, the new operating system is expected to include communications facilities, networking support, basic database facilities, multitasking, and demand-paged virtual memory.

Laser Printer Output Improving

More than one Japanese manufacturer is working on 1,200 dpi (dots-per-inch) laser printers with Postscript interfaces. These machines will produce quality output comparable to standard typesetting. Prototypes have already been shown in Japan with initial production expected to begin next year. And there are rumors of laser printers being developed that are capable of producing 2,000 dpi.

In the meantime, low-cost 450 dpi and 600 dpi laser printers with Postscript interfaces will begin to be shipped later this year. These will offer a significant increase in print quality and will no doubt lead to an increased use of laser printers for work previously done by typesetters. The introduction of the low-cost 450 and 600 dpi laser printers will no doubt cause 300 dpi machine prices to drop to well under $1,000. At the same time, the retail price for 300 dpi printers with Postscript and full-page graphics will probably drop to under $2,000.

PS/2 Sales Report

One noted market research group has reported that sales for PS/2 systems for the last quarter of 1987 broke down as follows:

Model 25	5 percent
Model 30	41 percent
Model 50	39 percent
Model 60	14 percent
Model 80	3 percent

This represents a dramatic increase in sales of the Model 50 compared to the Model 30, which represented most of IBM's PS/2 sales in the preceding quarter. The model 50 is IBM's lowest cost system capable of running OS/2.

There are many reports that IBM's PS/2 systems are being widely and heavily discounted by dealers and that most systems are readily available in the gray market. Discounts as high as 37 percent have been reported for the models 30 and 50 as dealers are attempting to move a large number of systems from inventory.

Good For The Gander

NEC recently sued **Seiko-Epson** for copying the BIOS in the NEC 286-based PC. They settled out of court, with Epson agreeing to pay an undisclosed sum to NEC and not to market the systems at issue.

It is worth noting that while NEC sued for ROM software copyright protection in a Japanese court, they argued in a U.S. court case, where they were being sued by Intel, that microcode stored in a microprocessor should not receive copyright protection.

NEC dominates the personal computer market in Japan with an estimated 80–90 percent of the 286-based market. □

PERISCOPE™ Periscope POWER

...Keeps you going full steam ahead when other debuggers let you down. With four models to pick from, you'll find a Periscope that has just the power you need.

. . . Start with the model that fits your current needs. If you need more horsepower, upgrade for the difference in price plus $10! And don't worry about having a lot more to learn . . . Even when you move to the most powerful model, Periscope III, an extra dozen commands are all that's involved.

Periscope's software is solid, comprehensive, and flexible. It helps you debug just about any kind of program you can write . . . thoroughly and efficiently. Periscope's hardware adds the power to solve the really tough debugging problems.

Periscope requires an IBM PC, XT, AT, or close compatible (Periscope III requires hardware as well as software compatibility); DOS 2.0 or later; 64K available memory; one disk drive; an 80-column monitor.

Periscope I includes a half-length board with 56K of write-protected RAM; break-out switch; software and manual for $345.

Periscope II includes break-out switch; software and manual for $175.

Periscope II-X includes software and manual (no hardware) for $145.

Periscope III includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 8 MHz is $995; for machines running up to 10 MHz, $1095.

Call Toll-Free for free information or to order your Periscope today!

MAJOR CREDIT CARDS ACCEPTED. **800-722-7006**

The PERISCOPE Company, Inc.

1197 PEACHTREE ST.
PLAZA LEVEL
ATLANTA, GA 30361
404/875-8080

VM/386

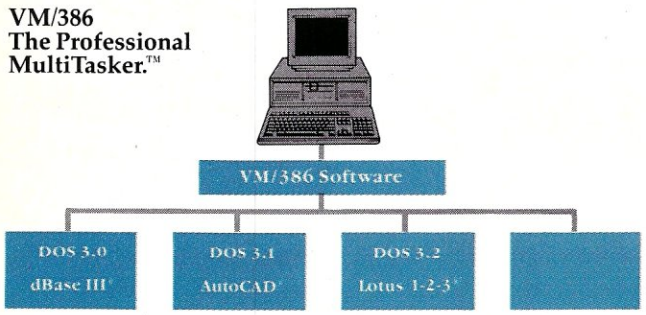
IT'S LIKE HAVING A DESKTOP FULL OF PCs

We know why you have it. Power. And now you can multiply that 386 power. Through true multitasking. With VM/386.™

VM/386 is the 80386 control program that brings you true DOS multitasking. VM/386 uses the virtual 8086 mode, built into the 80386 processor, to create individual virtual machines. You can load a different application in each virtual machine. It's like having a desktop full of PCs.

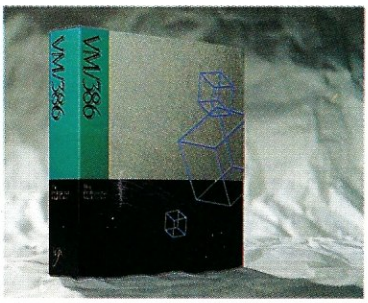
You have complete control over the virtual machines. You can tailor each virtual machine to fit your needs—and priorities. Each virtual machine has its own DOS, CONFIG.SYS, AUTOEXEC.BAT, and memory-resident programs along with its application. And each virtual machine is isolated from the others. A malfunction in one program doesn't destroy the others.

VM/386
The Professional
MultiTasker.™



There is virtually no limit to the number, or kind, of applications that can run simultaneously. For example, you can recalculate a 1-2-3® spreadsheet, sort a dBASE III® file, and receive your E-mail—all at the same time. You can even run more than one AutoCAD™ program simultaneously. And EGA applications run in the background as well as in the foreground—perfectly.

VM/386 protects your investment in software—and hardware. VM/386 will run all DOS software. Existing and future. Without modification. VM/386 is compatible with 80386-based PCs—either native or Intel® Inboard 386.™ VM/386 will support the monitors, hard disks, printers, mice, modems—even the network—you're currently using—or plan to use.



With VM/386 you use familiar commands. You don't have to buy new software or upgrades to get true multitasking. No PIF files. No special loaders. VM/386 works the way you do. Only lots faster.

VM/386 is easy to install. Easy to learn. And easy to use.

Dramatically increase your productivity. Start using all the power built into your 80386. Phone or write today.

IGC
4800 Great America Parkway
Santa Clara, California 95054
(408) 986-8373

System Requirements

80386-based computer such as COMPAQ® DESKPRO® 386® or 80286 computer with Intel® Inboard™ 386

One 1.2 Mb (5¼") disk drive or, one 3½" microfloppy

One hard disk drive

DOS 3.0 or later

2 Mb memory recommended

Not copy protected

Package includes both 5¼" and 3½" media.

VM/386 is a trademark of IGC.
AutoCAD is a trademark of Autodesk Inc.
COMPAQ and COMPAQ DESKPRO 386 are registered trademarks of Compaq Computer Corporation.
dBASE III is a registered trademark of Ashton-Tate Corporation.
Intel is a registered trademark of Intel Corporation.
Inboard is a trademark of Intel Corporation.
1-2-3 is a registered trademark of Lotus Development Corporation.

IGC

by Don Libes

Pointers to Functions

This column features tips and techniques for using the C programming language productively. Typical problems encountered in using C are discussed, and solutions offered. Readers who have suggestions, comments, and questions are encouraged to contact C Forum, Micro/Systems Journal, *Box 1192, Mountainside, NJ 07092.*

The last C Forum, "Dump from the Hip" (*Micro/Systems Journal*, February 1988) offered a simple tool that was able to take input from either memory or a file. In order to do that cleanly, we used "function parameters."

In this column, I am going to discuss function parameters in more detail. First, I will point out the most common mistake made in trying to pass functions as parameters. Namely, functions cannot be passed, only "pointers to functions" can be. Thus, the declaration

```
subr(fun)
 int fun();
```

will not work. The declaration of *fun()* says that it is a function returning an *int*, but we are not allowed to pass functions as parameters (my compiler says, "Warning: a function is passed as an argument").

What we want is to pass a pointer to a function.

```
subr(fun)
 int (* fun)();
```

fun() is now declared as "a pointer to a function returning an *int*." This is read "inside out" (Listing 1).

More complex function declarations are constructed in the same fashion. *cdecl*, a program that converts declarations from C to English, and vice versa, may also help (see "C Forum," *Micro/Systems Journal*, January/February 1987).

When calling, we supply *subr()* with a parameter simply by naming a function. The compiler arranges for the address of the function to be passed when it sees a function name; the & is not necessary.

To use the parameter while inside of *subr()*, we dereference *fun* using *, then treat the result as a function. In other words, give it a set of parentheses with arguments. For example, to call the function (passed to *subr()*) with arguments 17 and "hello world" we would say

```
foo = (*fun)(17,"hello
    world");
```

Notice that the usage looks just like the declaration we made earlier.

Last month's column presented a program doing exactly what we have just discussed. The function *real_dump()* was defined, which took a function parameter. *real_dump()* was declared as shown in Listing 2. This allowed us to read input from either a file or memory, depending entirely upon the function passed in *infunc*. *real_dump()* called *(*infunc)()* each time it needed the next input character. In a multitasking system, *infunc* could also return input from a co-routine, that is, another process executing concurrently. Once again, *real_dump()* would not need to be changed.

Sorting Anything

In *The C Programming Language*[1], authors Kernighan and Ritchie define a function named *sort()* as a function that has encapsulated the process of sorting so that it can sort any datatype. This includes integers, floats, strings, and any set on which you can define an ordering, such as dates and complex numbers.

The trick is that you, the caller, define the ordering on the datatype and pass a comparison function to *sort()*. Your ordering function will be called with two arguments of the datatype to be sorted. If the first is "larger" (in whatever sense you decide), your function should return 1. If it is smaller, it returns −1; and equal, returns 0.

To sort integers we can define *int_order()* as:

```
int int_order(x,y)
int x, y;
{
if (x>y) return(1);
if (x<y) return(-1);
return(0);
}
```

To sort strings, we can define *string_order()* as:

```
int string_order(x,y)
char *x, *y;
{
return(strcmp(x,y));
}
```

sort() actually requires two function parameters. The second one is used to swap two elements. To be efficient, *sort()* works on an array and moves pointers to the data being sorted rather than data itself. If we were sorting complex numbers, we would need a function to swap them. Here is a function to swap pointers to complex numbers (assume that COMPLEX is an appropriate *typedef*):

```
int complex_swap(x,y)
 COMPLEX **x, **y;
```

---

[1] Brian Kernighan and Dennis M. Ritchie, *The C Programming Language*, (New York: Prentice-Hall, 1978), p. 116.

10 Important Reasons C Programmers Use Our File Manager

1. It's written in C.
Clearly the growing language of choice for applications that are fast, portable and efficient. All of db_VISTA's source code is written in C.

2. It's fast – almost 3 times faster than a leading competitor.
Fast access that comes from the unique combination of the B-tree indexing method and the "network" or direct "set" relationships between records. A winning combination for fast performance.

3. It's flexible.
Because of db_VISTA's combination of access methods, you can program to your application needs with ultimate design flexibility. Use db_VISTA as an ISAM file manager or to design database applications. You decide how to optimize run-time performance. No other tool gives you this flexibility without sacrificing performance.
db_VISTA is also well behaved to work with most any other C libraries!

4. It's portable.
db_VISTA operates on most popular computers and operating systems like UNIX, MS-DOS and VMS. You can write applications for micros, minis, or even mainframes.

5. Complete Source Code available.
We make our entire C Source Code available so you can optimize performance or port to new environments yourself.

6. It uses space efficiently.
db_VISTA lets you precisely define relationships to minimize redundant data. It is non-RAM resident; only those functions necessary for operation become part of the run-time program.

7. Royalty free run-time.
Whether you're developing applications for yourself or for thousands, you pay for db_VISTA or db_QUERY only once. If you currently pay royalties to someone else for your hard work, isn't it time you switched to royalty-free db_VISTA?

db_VISTA™

Features
- **Multi-user** support allows flexibility to run on local area networks
- **File structure** is based on the B-tree indexing method
- **Transaction processing** assures multi-user consistency
- **File locking** support provides read and write locks
- **SQL-based db_QUERY** is linkable
- **File transfer** utilities included for ASCII, dBASE optional
- **Royalty-free** run-time distribution
- **Source Code** available
- **Data Definition Language** for specifying the content and organization of your files
- **Interactive database access** utility
- **Database consistency check** utility

File Management Record and File Sizes
- Maximum record length limited only by accessible RAM
- Maximum records per file is 16,777,215
- Maximum file size limited only by available disk storage
- Maximum of 256 index and data files
- Key length maximum 246 bytes
- No limit on number of key fields per record
- No limit on maximum number of fields per record

Operating System & Compiler Support
- **Operating systems:** MS-DOS, UNIX, XENIX, ULTRIX, Microport, VMS, Macintosh
- **C compilers:** Lattice, Microsoft, IBM, Aztec, Turbo C, XENIX, UNIX and LightspeedC

8. db_QUERY & db_REVISE.
Add the SQL-based, ad hoc query and report writer for a relational view of db_VISTA databases.
Use db_REVISE to re-design your database easily and quickly!
Both royalty free!

9. Free tech support.
60 days of free technical and application development support for every Raima product. Of course, extended support and training classes are also available at your place or ours.

10. Upward database compatibility
Start out with file management in a single-user PC environment—then move up to a multi-user LAN or a VAX database application with millions of records. You'll still be using db_VISTA. That's why so many C programmers are choosing db_VISTA.

John, You forgot one...
11. db_Vista training class
All you need to know to get the most from db_Vista! 2-5 days...Basics, Advanced, and Internals.
Call Now! Will cut development costs a lot!

30-day Money Back Guarantee!
Try db_VISTA in your environment for 30 days and prove it to yourself. If not completely satisfied, return it for a

Price Schedule

	db_VISTA	db_QUERY
☐ Single user	$ 195	$ 195
☐ Single user w/Source	$ 495	$ 495
☐ Multi-user	$ 495	$ 495
☐ Multi-user w/Source	$ 990	$ 990
NEW:		
☐ VAX Multi-user	$ 990	$ 990
☐ VAX Multi-user w/Source	$1980	$1980

Order Now.
Put db_VISTA to work in your application program. Ordering is easy—simply call toll-free. We'll answer your technical questions and get you started. Call today.

Call Toll-Free Today!
1 (800) db-RAIMA
(800/327-2462) or
206/828-4636

MasterCard VISA

RAIMA™ CORPORATION

3055 - 112th NE, Bellevue, WA 98004 USA
(206) 828-4636 Telex: 6503018237 MCI UW

MSJ 388

8-BIT CARDS
CHOKING
YOUR NETWORK'S THROUGHPUT?

Here's the solution!

16-Bit ARC-CARD/AT™ For AT COMPUTERS
16-Bit ARC-CARD/MC™ For PS/2™ Micro Channel™ Computers

One of the knottiest problems in network configuration is finding the right boards to maximize throughput. Thomas-Conrad's 16-bit cards make the difference.

- ***Highest performance*** **available for ARCNET® LANs**
- ***True*** **16-Bit data bus interface**
- ***Up to 50% faster*** **than 8-bit boards**
- ***Works with*** **EGA, EMS, 3278/79 Emulation Adapters. . . all in the same workstation or file server**

100% Burn-In, 2-Year Warranty

THOMAS-CONRAD CORPORATION
Connecting to the Future
8403 Cross Park Drive, Suite 1C, Austin, Texas 78754

800-332-8683
(In Austin, Texas, call 836-1935)

ARCNET is a registered trademark of Datapoint Corp. PS/2 Micro Channel are trademarks of International Business Machines Corporation.


```
{
 COMPLEX *temp;

 temp = *x;
 *x = *y;
 *y = temp;
 }
```

To sort an array of pointers to complex entities, we would then call:

```
sort(array,length,
     complex_order,
     complex_sort);
```

Condition Handlers

Another use of function pointers is for signal handlers. Condition handlers are functions called whenever some condition occurs. For example, division by 0 generates an arithmetic signal. Rather than aborting the program, we may transfer control to an error recovery routine, which, at the very least, can print out an error message. While debugging, it can trap back to the debugger.

Following the ANSI X3J11 draft C standard, we set up the signal han-

Listing 1

```
     (* fun)      "a pointer"
     (* fun) ()   "      to a function"
int  (* fun) ()   "               returning an int"
```

Listing 2

```
real_dump(address,infunc,ofp)
...
int (*infunc)();
```

Listing 3

```
#include <signal.h>

void sigfpe();  /* function to handle arithmetic
                   errors */

main()
{
  signal(SIGFPE,sigfpe);
  .
  .
}
```

Listing 4

```
#include <setjmp.h>
#include <signal.h>

void sigfpe();

main()
{
  jmp_buf env;
  .
  .
  if (SIG_ERR == signal(SIGFPE,sigfpe))
      perror("signal")
      exit(-1);
  }

  if (0 == setjmp(env)) {
      printf("inverse = %f\n",matrix_inverse(mat));
  } else {
      printf("no inverse\n");
  }
  .
  .
}
```

Listing 5

```
typedef int FUNC();          FUNC is a function
                             returning an int
typedef FUNC *FUNC_PTR;      FUNC_PTR is a pointer
                             to a FUNC
FUNC_PTR x[];                x is an array of
                             FUNC_PTRs.
```

Quit Wasting Time!

As a programmer, most of your time is spent writing and debugging source code, and documenting your work. A powerful, easy-to-use programmable text editor could be saving you HOURS of unnecessary effort.

Only MULTI-EDIT has all these time-saving features:

Fully automatic Windowing and Virtual Memory.
- Edit multiple files regardless of physical memory size.
- Easy cut-and-paste between files.
- View different parts of the same file.

Powerful, EASY-TO-READ high-level macro language.
- Standard language syntax.
- Full access to ALL Editor functions.
- Automate repetitive tasks.
- Easy, automatic recording of keystrokes.

Language-specific macros for ALL major languages.
- Smart indenting.
- Smart brace/parenthesis/block checking.
- Template editing.
- Supports C, Pascal, BASIC and Assembler.

Terrific word-processing features for all your documentation needs.
- Intelligent word-wrap.
- Automatic pagination.
- Full print formatting with justification, bold type, underlining and centering.
- Even a table of contents generator.

Compile within the editor.
- Automatically positions cursor at errors.
- Built-in MAKE capabilities.
- Run compiled program without leaving editor.
- Automatically allocates all available memory to compiler or program.

Complete DOS Shell.
- Scrollable directory listing.
- Copy, Delete and Load multiple files with one command.
- Background file printing.

Regular expression search and translate.
Condensed Mode display, for easy viewing of your program structure.
Pop-up FULL-FUNCTION Programmer's Calculator and ASCII chart.

and MOST IMPORTANT,
the BEST user-interface on the market!

- Extensive context-sensitive help.
- Choice of full menu system or logical function key layout.
- Function keys are always labeled on screen (no guessing required!).
- Excellent online, interactive tutorial.
- Keyboard may be easily reconfigured and re-labeled.

Users of Wordstar and Turbo Pascal's Editor could be programming in a fraction of the time with these features.

NO EDITOR ON THE MARKET TODAY HAS ALL THESE FEATURES, OR OFFERS YOU THIS MUCH POWER AT A REASONABLE PRICE, EXCEPT

Multi-Edit VERSION 2.0 $99.

COMPLETE
Or Get our FULLY FUNCTIONAL DEMO Copy for only $10!

	Multi-Edit	BRIEF 2.0	Norton Editor	Vedit Plus
Edit 20+ files larger than memory	Yes	Yes	No	Yes
Powerful high level macro language	Yes	Yes	No	Yes
Full UNDO	Yes	Yes	No	No
Visual marking of blocks	Yes	Yes	Yes	No
Column oriented block operations	Yes	Yes	No	No
Automatic file save	Yes	Yes	No	No
Online help	Extensive	Limited	Limited	Limited
Online tutorial	Yes	No	No	Yes
Choice of keystroke commands or menu system	Yes	No	No	Yes
Function Key assignments labeled on screen	Yes	No	No	No
WP Functions	Extensive	Limited	Limited	Extra Cost
Complete DOS shell	Yes	No	No	No
Pop-up Programmer's Calculator and ASCII table	Yes	No	No	No ASCII
Unlimited 'Off the Cuff' keystroke macros	Yes	No	No	Yes
Allocates all available memory to compiler when run from within editor	Yes	No	Yes	Yes
Intelligent indenting, template editing and brace/parenthesis/block matching and checking for all major languages	Yes	C Only	No	Limited
Flexible condensed mode display	Yes	No	Yes	No
Optional background communications and Spell Checker modules	Yes	No	No	No
PRICE	**$99**	**$195**	**$50**	**$185**

Requires IBM/PC/XT/AT/PS2 or full compatible, 256K RAM, PC/MS-DOS 2.0 or later-Multi-Edit and American Cybernetics are trademarks of American Cybernetics. BRIEF is a trademark of Underware, Inc. Norton Editor is a trademark of Peter Norton Computing, Inc. Vedit is a registered trademark of CompuView Products Inc. Copyright 1987 by American Cybernetics.

Get our FULLY FUNCTIONAL DEMO Copy for only $10!

To Order, Call 24 hours a day:
1-800-221-9280 Ext. 951
In Arizona: 1-602-890-1166
Credit Card and COD orders accepted

American Cybernetics
138 Madrid Plaza
Mesa, AZ 85201

dler as shown in Listing 3.

The first argument of *signal()* designates the condition to catch. Some other signals are SIGABRT (abnormal termination), SIGILL (bad or illegal function), SIGINT (interrupt from keyboard), SIGSEGV (memory access violation) and SIGTERM (normal termination). Many systems will also have extra operating system-dependent or machine-dependent signals.

The second argument to *signal()* is the function to execute upon receipt of the signal. It is also possible to pass several predefined values in place of a user-declared function. SIG_IGN means that a given signal should be ignored. SIG_DFL means that the behavior of a signal should be restored to what it was at program startup.

Here is an example of how we might use signals in a real application. First, let us assume that *matrix_inverse()* computes the inverse of a matrix. If the matrix is singular, i.e., has no inverse, a mathematical exception such as division by 0 will take place, which will cause our signal handler to be called (Listing 4). And here is the condition handler.

```
void sigfpe() {
longjmp(1);
}
```

The first call to *signal()* registers the signal handler. If *signal()* does not like its arguments, it returns SIG_ERR, for which we duly check. Otherwise, *signal()* returns a pointer to whatever function was last registered for the condition.

Next, we save our context using *setjmp()*. (See "C Forum," *Micro/Systems Journal*, January 1988, for more on *setjmp()*.) Initially, *setjmp()* returns 0 so that we proceed to call *matrix_inverse()*. If the matrix inverse is computed successfully, we continue after the else clause.

If *matrix_inverse()* raised any sort of arithmetic condition, such as would happen upon singular matrices, a SIGFPE would occur calling *sigfpe()*. *sigfpe()* executes *longjmp()*, which returns to the *setjmp()* and enters the else condition, printing "no inverse."

Final Notes

Earlier I demonstrated how to read function declarations "inside out." I also noted that *cdecl* can help you understand them. A final suggestion is the use of *typedefs*. Here is an example that will help explain:

Suppose we want to declare an array of functions. Or as I said before, we really mean "an array of function pointers," and furthermore, we must know the type that the function returns. In this case, let's say they return *ints*. Now we set up *typedefs*, starting with the functions and working up (Listing 5). Thus, x is clearly an array of function pointers.

By the way, to call the nth function stored in x[ ], we would say

```
result = (*x[n])(args);
```

Conclusion

I have discussed passing functions as parameters and demonstrated several uses. If we had written the examples without function parameters we would have forced the routines to know details immaterial to their operation. They would have to be modified each time we wanted to apply them to a new area.

By separating functions in this manner, we separate concerns. Functions are smaller, applicable to more situations, and become easier to read and debug. Algorithms are data independent—(e.g., *sort()*), I/O independent (e.g., *real_dump*) and algorithm independent (e.g., *signal()*). □

Acknowledgements: I thank Phil Nanzetta and Dave Oskard for proofreading many of these "C Forums."

Don Libes is a computer scientist working in the Washington, D.C., area on artificial intelligence in robot control systems.

dONEtwoTHREE™ dBASE® to 1-2-3® Gateway

is a dBASE add-on that gives you these amazing capabilities without leaving dBASE!

- Move between databases and worksheets with no intermediate files. Create new files or add to existing ones. Go both ways: dBASE to 1-2-3 and 1-2-3 to dBASE.
- Break up large databases into several, smaller worksheets.
- Instantly move single cells into dBASE variables and vice versa. Address cells by name or column-row.
- Browse worksheet in a window—horizontally, like 1-2-3, or with our unique, vertical browse.
- Write dBASE programs that use most of 1-2-3's powerful statistical, financial and data table functions, such as STD, VAR, AVG, NPV, FV, PV, etc.
- and much, much, more.

With **dONEtwoTHREE** access WKS, WK1 and WR1 files. And with our easy-to-use commands, create dBASE programs that integrate worksheets and databases into sophisticated applications. Includes an Assist-like utility to get even the novice started quickly.

dONEtwoTHREE is available in two editions: dBASE III Plus (also for FoxBASE and DBXL) or Clipper (also for QuickSilver). Coming soon: Editions for C.

dONEtwoTHREE is ridiculously underpriced at **$99.*** Comes with a 30 day money back guarantee, so you can't lose. Act now before the prices increase.

Communication Horizons • 701 Seventh Ave., #900 • New York, NY 10036
Tel: (212) 724-0150 Telex: 990868 Source: NAN285

*Add $5 for S&H. Add $3 for COD. MC or VISA accepted.

THE BEST OF BOTH WORLDS

Developing an application used to be easy — all you had to do was program it. But today, with countless languages, compilers, libraries, databases, editors, debuggers, and other tools, it is choosing the right development software that creates the real problem.

Now *The Andsor Collection* introduces a unique solution: a collection of sophisticated development tools, which you can use on their own, or together with your old ones.

The Andsor Collection is, of course, the superb application development system that programmers, VARs, and other developers have been using for over two years. And starting with Version 2.2, *The Andsor Collection* has acquired a new dimension: now you can access all its functions from within another program!

Think of it as a comprehensive, universal, language independent library. But *The Andsor Collection* is not a collection of subroutines: it is a seamless, integrated, interactive environment, specifically designed to expedite application development.

Whether you use C, Pascal, Cobol, Fortran, Basic, or any other language, *The Andsor Collection* can enhance your applications dramatically. Whether you add functions to an old program, or you write a new one, you can make them faster, more efficient, and more appealing.

Use *The Andsor Collection* to implement an entire application, or just portions of an application. You can, for example, create a windowed environment, add attractive data entry functions, define indexed data file structures, produce sophisticated reports or forms, and so on.

Although *The Andsor Collection* has far more features than other development systems, it is only one tenth their size. So the entire system can stay in memory, keeping all functions instantly accessible.

And *The Andsor Collection* is famous for its unique interactive environment. There is no conversion or translation — modify a procedure, a file definition or relation, a data entry screen, or anything else, and the change takes effect immediately, even while the application is running! Application development is a new experience.

The application users will benefit too. *The Andsor Collection* is amazingly fast, and since all data is in variable length format, the files take a fraction of the space needed with other systems. So not only will you develop your applications sooner, but they will be more efficient too. Whether you use *The Andsor Collection* alone, or to enhance a program.

So get the best of both worlds. Order *The Andsor Collection* today, and discover a whole new environment, without giving up your old development tools or your existing applications. Moreover, *The Andsor Collection* will be useful with all your future applications and languages.

You won't find a better value in development software: one program that is both a powerful stand-alone application development system, and a unique language independent collection of software tools; plus the run-time interpreter with unlimited royalty-free distribution. All for an incredibly low price. And with our 60 day money back guarantee*, you have little to lose and a lot to gain.

System Features

The Andsor Collection is the most versatile application development environment. And when using it with your programs, all its countless features can become part of your application. Hundreds of commands, functions, and options, are available to help you implement any application.

No list can be complete, so here are just some of these features: powerful database functions, maintenance-free multi-index data files, variable length data fields, unlimited file relations, complete window management, unique text processing functions, flexible data tables, powerful inquiry and reporting functions, versatile data entry capabilities, flexible procedural language, automatic error handling, extensive computational capabilities, data analysis and statistics, unique programmable charts, many printing functions, data communications, convenient system log file, full control over color attributes. And much, much more.

How It Works

This is simple and ingenious. Your program and *The Andsor Collection* reside in memory together (you load *The Andsor Collection*, which then loads your program). To transfer control to those portions of the application implemented in *The Andsor Collection*, you simply issue a software interrupt in your program, exactly the way DOS and BIOS functions are called. (If you are not familiar with this, examples in the manual show you how to do it.)

While in *The Andsor Collection*, the operation is identical to its operation when used alone. Finally, one command returns control to your program. And if you need this, simple commands transfer data directly to and from memory areas in your program (a number of formats are possible, all in standard ASCII, compatible with any language). Both *The Andsor Collection* and your program run as ordinary DOS programs. And you can also use them with other software (such as permanently resident programs).

"We looked at several systems and decided to standardize on The Andsor Collection. We are using it in many applications in several departments."

Gordon Eyers, Director of Information Services, Public Utilities Commission, Brantford, Ontario

"With The Andsor Collection we have achieved faster development and more efficient applications, which is important in large and complex projects like our Court Management System."

Dr. Mark Schrager, Consultant, Municipal Computer Services, Rochester, New York

"The Andsor Collection is unequalled when we need a solution in a hurry. Applications that we have implemented include modeling, data acquisition and analysis, and reporting."

Joe Blask, Engineering Support, General Motors, Troy, Michigan

ANDSOR® — *The Andsor Collection™*

ANDSOR RESEARCH INC.
390 Bay Street, Suite 2000
Toronto, Ontario M5H 2Y2
(416) 245-8073

To order call toll free
(U.S. and Canada)

1-800-628-2828
Ext. 535

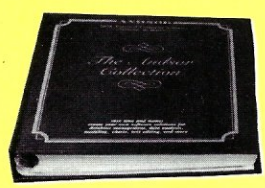
$145 *(includes shipping*)*

Visa, MC, AmEx, Check

*Price includes shipping in the U.S. and Canada. Please add $10 for shipping to other countries. If you return the software, $8 will be deducted from the refund, to cover our shipping cost.

System requirements: any IBM PC or PS/2 or fully compatible, 250K+ (excluding DOS and other programs), one disk drive or hard disk, monochrome or color monitor, DOS 2.0+ or OS/2

© 1988 Andsor Research Inc. Andsor is a registered trademark and The Andsor Collection is a trademark of Andsor Research Inc.
IBM is a registered trademark and IBM PC, PS/2, OS/2 are trademarks of IBM Corporation



4 TIMES FASTER THAN TODAY'S FASTEST ASSEMBLER!

That's right. 4 times faster.

Clocking in at over 75,000 lines per minute on a 6MHz IBM AT, OPTASM is four times faster than Microsoft's MASM 5.0. **4 times faster** — that's 400% more throughput!

But speed is only one part of it. OPTASM is nearly 100% compatible with MASM 5.0 (except 386 support).

It is the only single assembler capable of supporting the various incompatibilities between MASM 3, 4 & 5. That makes OPTASM **more MASM compatible than any single version of MASM!**

Other features? OPTASM generates smaller code without ever generating extra NOP's. It automatically handles jumps out of range, up to 15,000 symbols and most of MASM's phase errors. It also boasts a built in MAKE and simplifies segmentation.

That's why we can make our OPTASM challenge: Test OPTASM head to head against MICROSOFT MASM 5.0. Order **both** assemblers with their 30-day guarantees. In a lot less than 30 days, you'll see just how dazzling OPTASM's speed really is. You'll realize that we're compatible, easier to use, and deliver many more important features than MASM. So accept our challenge. Try both assemblers. **Four times faster and more features, too.** We know which one you'll send back.

Write or call us to order or for our detailed brochure.

OPTASM: $195 Guaranteed returnable within 30 days.

WHAT DO PROGRAMMERS SAY ABOUT OPTASM?

"It (OPTASM) just blows MASM away . . . reduces my assemble time for Periscope from 3-plus minutes to less than 45 seconds."
Brett Salter, President,
The Periscope Company

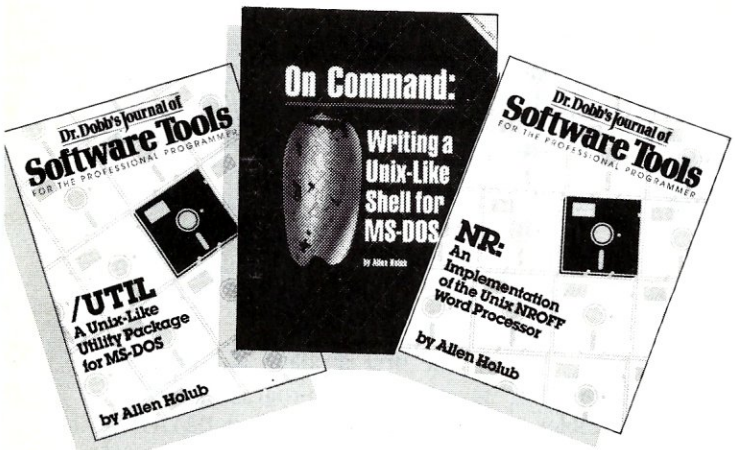
"OPTASM has been absolutely solid. For me, the most useful new product in 1987."
Chris Dunford, Columbia, MD

SYSTEMS

1622 N. Main Street
Butler, PA 16001
412-282-0864
BBS 412-282-2799
Telex 559215
800-833-3061

Bring the Conveniences of UNIX To YOUR MS-DOS Machine

Full Source Code on Disk!



On Command: Writing a Unix-Like Shell for MS-DOS

by Allen Holub

This book and ready-to-use program demonstrate how to write a Unix-like shell for MS-DOS, with techniques applicable to most other programming languages as well. The book and disk include a detailed description and working version of the Shell, complete C source code, a thorough discussion of low-level MS-DOS interfacing, and significant examples of C programming at the system level.

Supported features include: read, aliases, history, redirection and pipes, Unix-like command syntax, MS-DOS-compatible prompt support, C-Shell-based shell scripts, and a Shell variable that expands to the contents of a file so a program can produce text that is used by Shell scripts. The Unix-like control flow includes: if/then/else; while; foreach; switch/case; break; continue.

The ready-to-use program and all C source code are included on disk. For IBM PC and direct compatibles.

Book & Disk (MS-DOS) **Item #29-1** **$39.95**

\Util

by Allen Holub

When used with the Shell, this collection of utility programs and subroutines provides you with a fully functional subset of the Unix environment. Many of the utilities may also be used independently. You'll find executable versions of cat; cp; date; du; echo; grep; ls; mkdir; mv; p; pause; printevn; rm; rmdir; sub; and chmod.

The \Util package includes complete source code on disk and all programs (and most of the utility subroutines) are fully documented in a Unix-style manual. For IBM PC and direct compatibles.

Manual & Disk (MS-DOS) **Item #12-7** **$29.95**

NR: An Implementation of the Unix NROFF Word Processor

by Allen Holub

NR is a text formatter that is written in C and compatible with UNIX's NROFF. Complete source code is included in the **NR** package so that it can be easily customized to fit your needs. **NR** also includes an implementation of the -ms (manuscript) macro package and an in-depth description of how -ms works. **NR** does hyphenation and simple proportional spacing. It supports automatic table of contents and index generation, automatic footnotes and endnotes, italics, boldface, overstriking, underlining, and left and right margin adjustment. **NR** also contains:

- extensive macro and string capability
- number registers in various formats, including Roman and Arabic numerals, both spelled out and in outline form
- diversions and diversion traps (macros that are triggered automatically)
- input and output line traps

NR is easily configurable for most printers. Both the ready-to-use program and full source code are included.

For PC compatibles.

Manual & Disk (MS-DOS) **Item #33-X** **$29.95**

Save 15%

Receive **On Command, Util** and **NR** together for only $85.95! You get all the convenience of Unix-like features and save 15%.

Unix-like Features Package **Item #167** **$85.95**

To Order:

Return this order form with your payment to:
M&T Books, 501 Galveston Dr., Redwood City, CA 94063
Or, **CALL TOLL-FREE 800-533-4372** Mon-Fri 8AM-5PM
(In CA call 800-356-2002)

ORDER FORM

Name _____

Address _____

City _____ State _____ Zip _____

Yes! I want to SAVE 15%! Please send me the

Unix-Like Features Package for only $85.95 _____

Send me **On Command** book & disk $39.95 _____

Util manual & disk $29.95 _____

NR manual & disk $29.95 _____

Subtotal _____

CA residents add sales tax ____% _____

Add $2.25 per item for shipping _____

TOTAL _____

☐ Check enclosed. Make payable to M&T Publishing.

Charge my ☐ VISA ☐ M/C ☐ Am. Ex.

Card No. _____ Exp. _____

40A

by Stephen R. Davis

Virtual Memory Techniques: Part 1—Arrays of pointers

This column features tips and techniques for using Turbo Pascal productively on MS-DOS/PC-DOS microcomputer systems, and discusses typical programming problems and their solutions. Reader suggestions, comments, and questions are encouraged, and can be addressed to "Turbo Pascal Corner," Route 5, Box 107K, Greenville, TX 75401, or through MCI mail, 289-6124.

I recently had the opportunity to speak with Brad Silverberg, the vice president of languages for Borland. I asked him what Borland's plans were concerning the further development of its languages. Here is a summary of what he had to say.

Borland's Plans

Borland is keenly aware of the lack of a debugger in the Turbo interface. They have announced a Turbo debugger, which should begin delivery during the first quarter of this year. Unfortunately, it will initially be a standalone product, but it should be integrated into the language environment very shortly. This is essential if Borland is to "keep up with the Microsofts" as this is a major failing in the Turbo environment when compared with the Quick languages.

With respect to Turbo Pascal 4.0, a limitation in the initial release was the lack of emulation support for the IEEE extended range floating point. Without an 8087, the 6-byte Real was the only format available. This effectively made the other real formats useless for commercial development. Borland promises complete emulation for these real formats very soon, perhaps by the time you read this. Apparently, this will be the exact same emulation software as in Turbo C.

At least Borland is considering the development of a Turbo Assembler. They are definitely adding assembly language capabilities to both Turbo C and Turbo Pascal to handle inline assembly code. No new Macintosh language products are planned, although continued support for Mac Pascal is promised.

One final note: Borland considers the Paradox database language to be another member of the language family. Hooks are being added to both Turbo C and Turbo Pascal to allow programs written in these languages to access Paradox. Conceptually, this is like Embedded SQL in languages such as Cobol (an especially accurate comparison since Borland will be adding SQL to Paradox this year). Such access opens a wealth of database power to the Turbo Pascal programmer.

This Month's Topic—Virtual Memory

Accessing large amounts of memory has always been a problem with Turbo Pascal. Turbo Version 3.0 and earlier generations belied their CP/M roots by only allowing the user to directly access 64K of RAM. While Version 4.0 has extended that restriction to the 640K limitation of DOS itself, this is still not nearly enough room for many programmers. This limitation is pretty silly when there is often 20 or more megabytes of hard disk storage space sitting in the same machine (or several megabytes of EMS memory). The problem is how to get at it.

The most straightforward approach is for the user to simply read and write data, as needed, to and from disk. For example, the user might have a few large buffers, named something generic, such as DATA1, DATA2, etc. He could then read data into these buffers as required so that at one point in time DATA1 might contain, say, social security numbers, while at a different time it would contain addresses. Although this plan would probably work, it suffers from a few serious problems.

The first problem is that at any given time the programmer cannot tell just by looking what DATA1 has in it. Giving DATA1 various pseudonyms via an ABSOLUTE declaration does not help. Debugging such a program is also very difficult.

The second problem is a matter of taste. Application routines should only have to worry about application problems—adding the logic to manage data on and off the disk obscures the program, which further complicates the debugging job.

Both of these objections can be addressed using a programming technique known as virtual memory. With virtual memory, the applications program simply declares its data structure to be of any size required, limited only by the amount of available disk space. Rather than access this memory directly, however, the user program must access it through an intermediate known as the virtual memory manager (VMM).

The VMM breaks up the declared data structure into smaller chunks. It is important that these chunks be small enough to fit several into memory at any given time, yet large enough that the VMM will not need to allocate them too frequently. Usu-

EVEN MORE POWER AND FLEXIBILITY

BRIEF 2.0

Users and industry press alike have unanimously proclaimed BRIEF as the best program editor available today. Now, the best gets better, with the release of BRIEF 2.0.

Straight from the box, BRIEF offers an exceptional range of features. Many users find that BRIEF is the only editor they'll ever need, with features like real, multi-level Undo, flexible windowing and unlimited file size. But BRIEF has tremendous hidden power in its exclusive macro language. With it, you can turn BRIEF into your own custom editor containing the commands and features you desire. It's fast and easy.

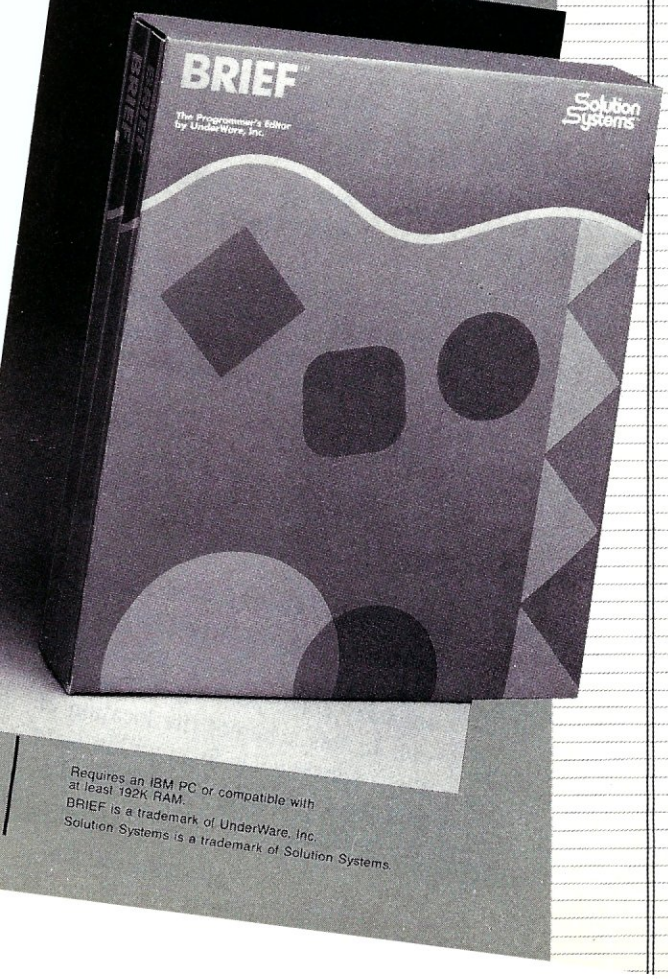
Jerry Pournelle, columnist for BYTE magazine summed it all up by saying BRIEF is, "**Recommended. If you need a general purpose PC programming editor, look no further.**" His point of view has been affirmed by rave reviews in C JOURNAL, COMPUTER LANGUAGE, DR. DOBB'S JOURNAL, DATA BASED ADVISOR, INFOWORLD AND PC MAGAZINE.

One user stated "BRIEF is one of the few pieces of software that I would dare call a masterpiece." Order BRIEF now and find out why. **BRIEF 2.0 is just $195. If you already own BRIEF, call for upgrade information.**

TO ORDER CALL: 1-800-821-2492
(in MA call 617-337-6963)

As always, BRIEF comes with a 30 day money-back satisfaction guarantee.

Solution Systems™
541 Main Street
Suite 410J
So. Weymouth, MA 02190
(617) 337-6963



Requires an IBM PC or compatible with at least 192K RAM.
BRIEF is a trademark of UnderWare, Inc.
Solution Systems is a trademark of Solution Systems.

Look at these BRIEF 2.0 enhancements!

Main Features:

- *All new documentation with tutorials on basic editing, regular expressions **and** the BRIEF Macro Language.*
- *Setup program for easy installation and configuration. (Requires no knowledge of the macro language)*
- *Increased speed for sophisticated operations like Undo and Regular Expression Search.*
- *Expanded regular expressions, with matching over line boundaries.*
- *More block types, with marking by character, line or column.*
- *Command line editing (move cursor, add and delete characters, specify command parameters).*
- *Support for more programming languages.*
- *Optional borderless windows.*
- *Enhanced large display support, including wider displays.*
- *Reconfigurable indenting for C files (supports most indenting styles).*

Plus the basic features that made BRIEF **SO** popular!

Basic Features:

- *Full multi-level Undo*
- *Windows*
- *Edit many files at once*
- *File size limited only by disk space*
- *Automatic language sensitive indentation*

ally, the nature of the data suggests a logical division. When the user's program asks for a block of data, the VMM checks to see if it isn't already in RAM. If it is, the VMM returns a pointer to where the block is located. If it is not, the VMM first reads the block from the disk. Reading a new block of data into RAM usually means that some older block of data must be removed from RAM to make room. The user program is not directly involved in the actions taken by the VMM to honor its request.

Of course, disks are *much* slower than RAM. Forcing the program to the disk on every access of its data structure could result in extremely slow execution. However, the VMM relies on the fact that, at any given time, a program's interest is usually isolated to a small number of particular areas of the data.

The user program does have to accommodate the VMM in this respect. If it will be necessary to search the data for particular values, then the data must be indexed (or hashed) using an index array that can comfortably fit entirely in RAM. The amount of "hopping around" in the data structure must be kept to a minimum to achieve acceptable performance. This hopping about translates to a large amount of disk thrashing; reading sectors from the disk so that the user program can take a single look at the data before going on to yet another disk request. For example, sorting data in virtual memory using a bubble sort, which makes multiple passes through the data, would lead to unacceptably long execution times.

Listing 1. Sorting matrices of data rapidly using an array of pointers as an index

```
type
     ptr = ^data;
     data = array [0..99] of integer;

var
     pdata : array [1..10] of ptr;
     ddata : array [1..10] of data;
     i,j   : integer;
     temp  : ptr;
     flag  : boolean;

begin
     {print explanation of output}
     writeln ('Data first in physical and then in sorted order');
     write ('Col 1 is the sort column');
     write ('  ---  remaining columns contain the row number');

     {initialize the matrix to predictable values - make col 0
      the sort field}
     for i := 1 to 10 do
     begin
          ddata [i][0] := random (100);
          for j := 1 to 99 do
               ddata [i][j] := i
     end;

     {initialize 'pdata' to point to the idividual arrays}
     for i := 1 to 10 do
         pdata [i] := addr (ddata [i]);

     {now perform a bubble sort on column 0 of 'ddata' - notice
      how we need only exchange the pointer rather than swap the
      entire arrays of 100 integers}
     repeat
          flag := true;
          for i := 1 to 9 do
               if (pdata [i]^[0] > pdata [i+1]^[0]) then
               begin
                    temp        := pdata [i+1];
                    pdata [i+1] := pdata [i];
                    pdata [i]   := temp;
                    flag := false
               end
     until flag;

     {now print out 'ddata' both sorted and unsorted}
     for i := 1 to 10 do
     begin
          writeln;
          for j := 0 to 15 do
             write (ddata [i][j]:4)
     end;
     writeln;
     for i := 1 to 10 do
     begin
          writeln;
          for j := 0 to 15 do
             write (pdata [i]^[j]:4)
     end
```

Computer Memory Addressing

Before we write an example VMM, we should first review how a computer stores data structures in memory. The simplest such structure is the array. In an array, the elements are uniform and numbered, and the location of each element is a function of its number. The individual elements may themselves be composite (non-atomic) structures. Arrays of records and arrays of arrays, also known as matrices, are common. A reference to an element in an array is resolved by multiplying its number by the size of each element and adding this to the address of the base of the array. Take the following example:

```
type      subarray : array
          [1..40] of byte;
var
   data : array [1..80] of
          subarray;
```

In *data*, i.e., an array of an array of bytes, any particular byte is specified by providing two subscripts. A reference *data [i][j]* is resolved as follows:

```
location of data [i][j] =
  location of data + i *
  40 + j
```

In memory, the 40-byte subarray corresponding to *data [1]* appears first, followed by the second 40-byte subarray corresponding to *data [2]*, etc. In this scheme, the location of these subarrays is fixed within *data* and their space allocated is fixed at declaration.

This type of structure is too inflexible for our purposes. With a slight modification, however, we can make this scheme very flexible indeed. Consider the following definition:

```
type
    subarray = array
      [1..40] of byte;

var
    pdata : array [1..80]
      of ^subarray;
```

Here we have defined *pdata* as an array of pointers to arrays. Each *pdata[i]* contains not a subarray, but the address of a subarray. The individual bytes of such a declaration are accessed as *data[i]^[j],* the circumflex indicating indirection.

With such a declaration, there is no implicit ordering of the subarrays in memory. Each *pdata [i]* points to where the subarray is located. In fact, not all of the subarrays need be of the same length, or even present at all. In a technique known as "sparse matrices," those elements of *pdata* that contain no information are assigned the value NIL.

Let's take a look at a simple example application of this concept. It is often necessary to sort a matrix by one of its columns. If we stay strictly with the "array of arrays" approach, the exchanging elements necessary to perform the sort requires that we copy entire subarrays. This can become quite a performance problem if the subarrays become large.

We can avoid this problem by assigning an "array of pointers to arrays" to point to the individual rows of the matrix. We can now sort the matrix by merely exchanging pointers in this indexing array. Listing 1 shows just such a program which bubble-sorts a large matrix by the random data assigned to column 0. At the end of the program, the program prints out the matrix both in "physical" order and in sorted order.

Review the program until you understand the role of *pdata* fully. Notice in particular how *temp* is used to exchange two entire rows of the matrix with three simple assignments. Also notice that the speed of the sort is not a function of the length of the subarrays. Change the "0 . . 99" declaration of *data* into a "0 . . 9999" and notice that there is no difference in execution speed. Rewrite the bubble sort routine to sort *ddata* as a normal matrix and compare performance.

Of course, we can define arrays of pointers to things other than arrays. An array of pointers to records exhibits the same advantages as those outlined above. A record has a further advantage: its constituent members need not be of the same type. This will allow us the flexibility we need to place elements of our VMM in memory as desired, a topic we will continue to develop in our next installment. □

Stephen Randy Davis is one of M/SJ*'s technical editors and a systems programmer. He lives in Greenville, Texas, and is currently completing a Master's degree in Physics.*

Conquer Time and Space.

Introducing XO-SHELL™.
Pop-Up Productivity for Programmers.

No matter what language you program in, XO-SHELL will help you hurdle the barriers to working faster and more efficientiy by eliminating programming hassles. Only with RAM-resident XO-SHELL can you:

- DO CROSS-REFERENCING without leaving your editor
- VIEW ANY FILE and TRANSFER ANY SECTION into your editor or to your printer
- VIEW, COPY and ERASE files directly from a SCROLLABLE DIRECTORY DISPLAY
- With a single key stroke RETRIEVE previous DOS commands, then EDIT and REEXECUTE them
- DO SOURCE-LISTING while in your application
- OBTAIN KEY-CODES without a reference and without going through difficult interpretation
- INSERT GRAPHICS CHARACTERS in your source code.

XO-SHELL is for PCs, XTs, ATs, PS/2s, compatibles.

WYTE™

WYTE CORPORATION
701 Concord Avenue
Cambridge, MA 02138

$49
plus $5 shipping & handling
Call today toll-free
(800) 635-5011
In MA: (617) 868-7704
Visa, MasterCard

TP2C C CODE from Turbo Pascal
Power, Practicality, Portability from BISS

TP2C is an investment that pays for itself . . . by making a programmer's job quicker and easier. The TP2C translator converts Turbo Pascal source and produces standard C code. It translates whole programs, or program segments, and even handles nested procedures and inline assembly statements . . . all the while utilizing your valuable investment in Turbo Pascal.

Need to know more? TP2C allows you to write more powerful and flexible programs, is easily integrated with C programming, and allows easy adaptation to UNIX and other environments. TP2C is designed for PC-DOS and MS-DOS machines.

Sure, there's much more! Call **BISS** today. Angie or Rick will be happy to help you. Ask for our TP2C . . . at $249.

Business Information Systems and Services

* Turbo Pascal is a registered trademark of Borland International

BISS . . . 504-295-0260
5432 Superior Drive · Baton Rouge · LA 70816

DEBUGGING SWAT TEAM

Order Eco-C88 Rel. 4.0 New Modeling Compiler and get C-more at no extra charge!

Seek and Correct

You already know that fast compilation does not mean fast program development. Backing up for bogus error messages and removing the bugs takes time. Eco-C88's "Seek and Correct" three - way error checking finds even the most elusive bugs, clearing the path for swift program development.

Double Barrel Error Checking

Eco-C88 nails **syntax errors** cold and tells you about the error in plain English. And there's no avalanche of false error messages, either. Other compilers can generate up to four times the number of error messages actually present; they leave it up to you to guess which ones are real. You'll be more productive with Eco-C88 because there is no guess work.

Eco-C88 provides ten levels of **semantic error** checking. You can select from almost no checking to the fussiest you've ever seen. Eco-C88's "picky flag" finds subtle errors that slip by other compilers.

Eco-C88 also features:

- All data types, plus ANSI Enhancements
- Robust library, including many new ANSI functions
- CED editor with online function help, split windows, compile-edit-link capability
- New, expanded manual with sample programs for the library functions

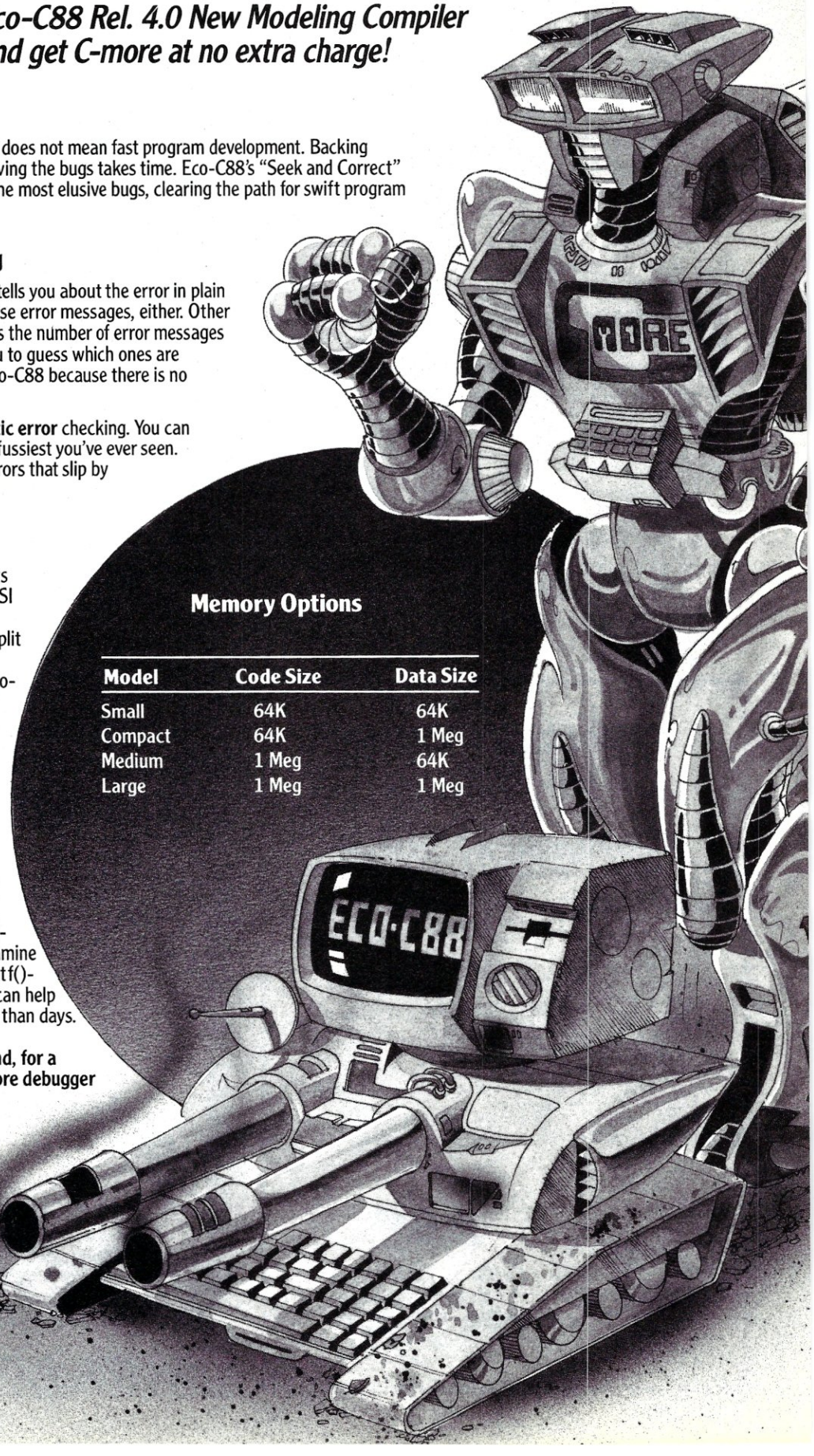
C-more Source Code Debugger

Finally, if a really nasty bug persists, put C-more, our source code debugger, to work. With C-more you can watch your program as it executes, single-step it, set simple or conditional breakpoints, test complex expressions, use variables as indexes into other variables, initialize and trace variables, examine CPU registers, display results with printf()-type options and much more. C-more can help you track down bugs in minutes rather than days.

The price for Eco-C88 is $99.95. And, for a limited time, we'll give you our C-more debugger at no extra charge.

Memory Options

Model	Code Size	Data Size
Small	64K	64K
Compact	64K	1 Meg
Medium	1 Meg	64K
Large	1 Meg	1 Meg



Ecosoft Inc.
6413 N. College Ave.
Indianapolis, IN 46220

(317) 255-6476 (Tech Info)
(800) 952-0472 (Orders)

Technical Brief

Scanning the OS/2 Horizon

by David E. Cortesi

There's a lot of talk about IBM's new Operating System/2. But what is OS/2, and how does it differ from the old familiar DOS? Being a multitasking operating system, OS/2 is capable of loading multiple independent applications and sharing the system hardware in such a fashion that all appear to run concurrently. OS/2 keeps the different applications from interfering with each other. OS/2 can even run one DOS application concurrently with other programs.

OS/2 needs an Intel 80286 or 80386 CPU and at least 2 MB of RAM, i.e., one of IBM's new Personal System/2 machines (model 50 or above), an IBM PC/AT or compatible, or one of the new 386 systems. Be careful choosing non-IBM hardware; OS/2 is a rigorous test of compatibility and won't boot on many "compatibles."

Share and Share Alike

OS/2 uses different methods in sharing each hardware resource: the CPU is time-shared among loaded programs; storage is segmented and allocated using the addressing hardware; and disks are managed as in an MS-DOS system with network-style file sharing. Three fundamental concepts govern resource allocation: the screen group, the process, and the thread.

The *screen group* represents the hardware used by the operator: the display screen, the keyboard, and the mouse. OS/2 supports only one set of hardware, but arranges things so there appear to be 16 virtual displays, each with its virtual mouse and keyboard.

A *process* represents a loaded program plus the resource it owns, such as code and data segments, open files, default drive and directory path, and environment strings.

The *thread* is the executable unit; the entity that is allowed to share the use of the CPU. If a process corresponds to a playing field, then a thread corresponds to the teams that play on it. A thread is actually just a set of registers that occupy and animate the static process.

A screen group will ordinarily be used by just one process, but a process can create other processes and these descendants can share the same screen group. Ordinarily, a process will be animated by only one thread, but one thread can create another so a program can have dozens of concurrent threads in its process (and the system limitation on threads is quite generous).

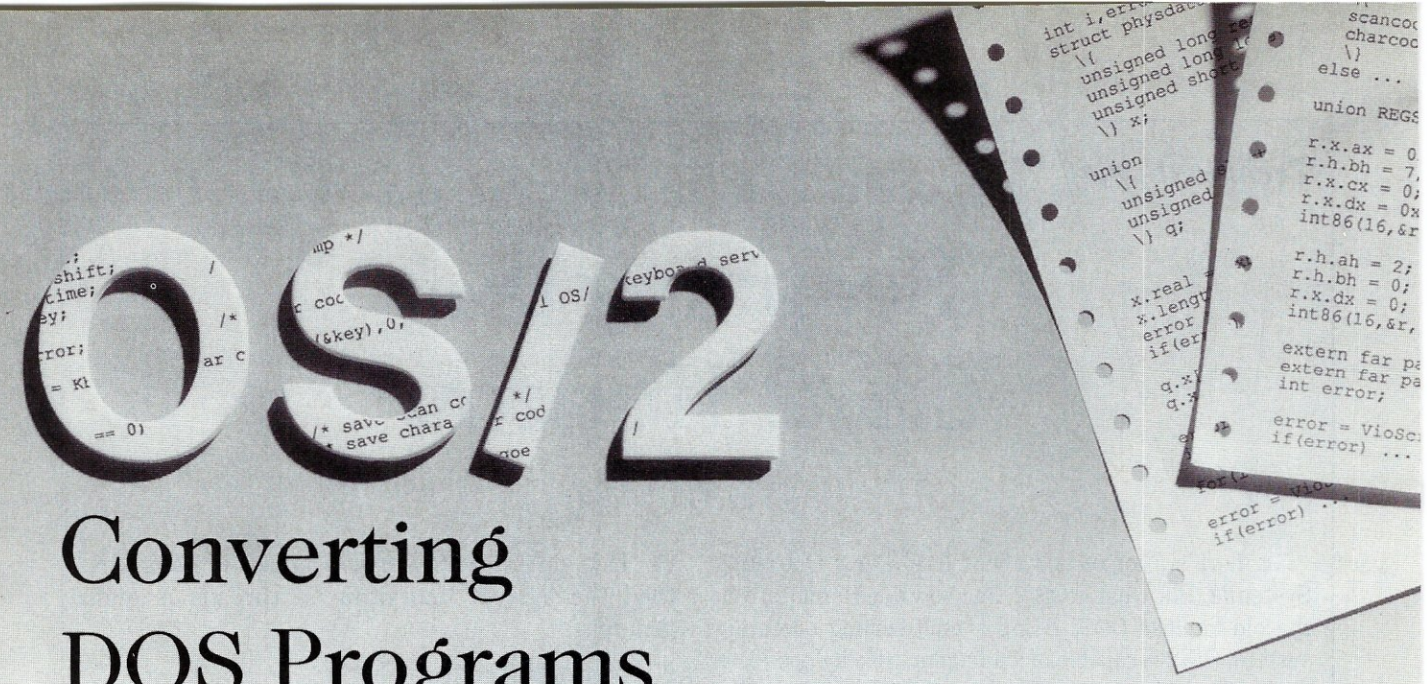
Life in Protected Mode

DOS operates only in the real mode of the 286 CPU. OS/2 takes advantage of what Intel calls the 286's protected mode of operation, and this mode has a lot to do with the character of OS/2 itself. In real mode the programs share one common megabyte of address space. In protected mode there's as much as 16 MB of storage, but all references to it are indirect via the entries in two descriptor tables prepared by the operating system, the Global Descriptor Table (GDT) and the program's own Local Descriptor Table (LDT). The operating system has total control over what the program may touch, modify, or execute. Attempts to access memory outside of that provided by the operating system generate a hardware exception.

Because of the protected mode, OS/2 can treat storage as a pool from which segments are allocated for different programs. The first 640K of storage are given over to a DOS process. The display buffer and ROM segments are set aside under privileged descriptors. Compatible programs can refer to these areas, since they fall into the address space of a real mode program. For DOS, a linked program is a binary image where every part of the program runs together into a sequence of bytes. For protected mode, the identities of segments must be preserved so that the LDT can be built to describe each segment as code or data.

In addition, OS/2 supports dynamic linking, in which some of a program's references to external procedures may be left unresolved until the program is loaded. The object of a dynamic link is an exported name in a dynamic link library. A dynamic link library is an executable file that is prepared like a program by compiling and linking. However, a dynamic link library is not an independent program, but only a package of subroutines that is exported to the programs that refer to them. OS/2 loads dynamic link libraries as required when it loads a client program that needs one. Since a dynamic link package is loaded like a program, it is free to refer to other dynamic

. . . *continued on page 34*



Converting DOS Programs to OS/2 Protected Mode

by David A. Schmitt

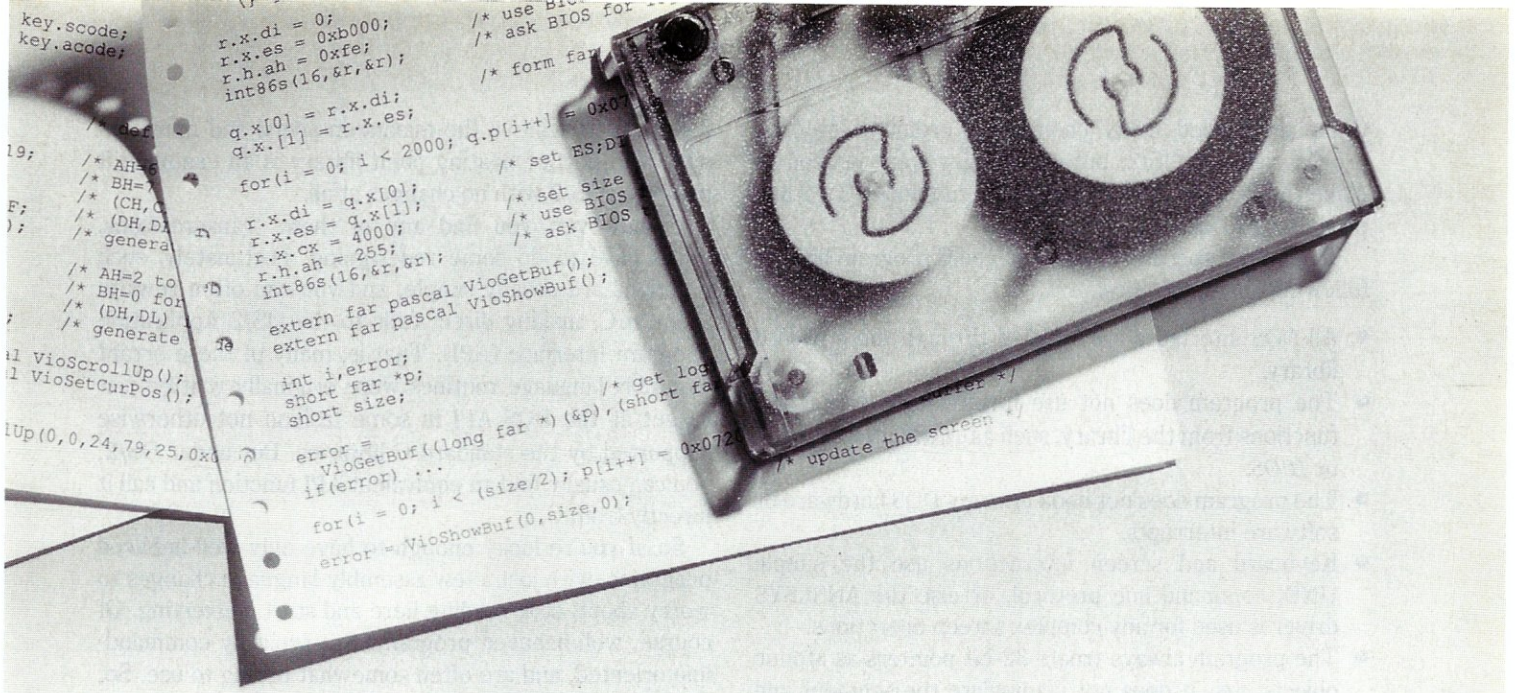
Porting DOS programs to OS/2 is really fairly simple, assuming you follow a few established guidelines.

My company, Lattice, Inc., has been working with pre-release copies of IBM's OS/2 Standard Edition 1.0 since June 1986. During that time, we've converted large amounts of C and assembly language code to run in the OS/2 protected mode, and I'd like to share some of our experiences with you.

Let me begin by saying that our relationship with IBM and OS/2 has been consistently very positive. We agreed to be a test site for OS/2 with some trepidation, since testing a new operating system is usually tedious and frustrating. However, IBM answered all of our questions promptly and accurately, and with each update, OS/2 has become better and better.

By April 1987, when IBM formally announced the availability of this new operating system, it was reliable enough to be routinely used in several of our programmers' workstations, which at that time were IBM PC/AT systems. The major deficiency that prevented us from switching completely to OS/2 was its lack of LAN support, which is a vital part of Lattice's software development environment.

The few remaining problems that we observed after the announcement date were all corrected when OS/2 was officially released in December, and we remain un-



abashed OS/2 boosters. I can't imagine a software developer not getting excited by this major improvement over DOS, and I think we're going to see some significant applications built on top of OS/2.

Let's take a look at the most immediate problem many programmers are now facing: Can I move my DOS software to OS/2 protected mode, and if so, how do I do it? We've already come up with some guidelines that are particularly appropriate for C and assembly language, but can probably be adapted for other languages as well.

Programmer, Know Thy Code!

The first few rules are rather obvious, but I've seen several OS/2 porting efforts derailed because one of these "obvious" steps was overlooked:

- Make sure you have all of the source modules that comprise your application. If you depend on somebody else's code and it's only available in object form, you'll probably be roadblocked. If the object code is a library product that you purchased without source code, you might be able to convince the publisher to give you the source code if you agree to send back the changes that you make for OS/2. On the other hand, if the library is really complex, you might try bribing the publisher so he'll do the work.
- Make sure you know how to build the application. Some of the executables (i.e., .EXE files) have been floating around for some time, and the folks who originally built them may have since disappeared without leaving any tracks.
- Go ahead and actually build the application under DOS to see if it still works. Use the latest version of the C compiler and assembler that you can, since old source code sometimes doesn't make it through the more recent language translators because of improvements in syntax checking.
- If the source code you're using produces a flawed executable, debug it under DOS since you probably work more efficiently in that environment.
- Once you have a clean version running under DOS, save the source and executable. You'll probably need to refer back to this baseline version when you evaluate the performance of the converted program running under OS/2.

Remember that porting to OS/2 is a lot like painting a room: If you do a good job of preparation, the actual work is much easier and the result usually turns out better.

Programs: The Good, The Bad, and The Ugly

The next step is *triage*, or classifying your application according to the likelihood that it will survive the operation. We've identified the following four classes:

1) Well-Behaved Programs

Well-behaved programs use very few operating system services except for the File Manager, which they access through the C library via functions such as *fopen*, *getchar*, *read*, and so on. Since OS/2 has essentially the same file system as DOS, the OS/2 C library can provide exactly the same services, which makes well-behaved programs really easy to convert. In fact, most of the time you can just run the source code through the OS/2 compiler or assembler and then build an executable with the OS/2 linker.

This sounds so simple that you probably wonder if there are any significant programs in this category. Actually there are quite a few. We found that most of our programming tools, e.g., compilers, assemblers, GREP, DIFF, and so on, are well-behaved and moved to OS/2 with few problems. Of course, the Lattice MS-DOS C

Compiler needed many changes in order to produce proper code for OS/2, but no changes were needed to make the original MS-DOS compiler run under OS/2 and produce MS-DOS code.

To be more specific, a well-behaved C program has the following characteristics:

- All DOS interfaces are handled through the standard library.
- The program does not use low-level DOS interface functions from the library, such as *int86, intdos, bdos,* or *BIOS.*
- The program does not hook into any DOS hardware or software interrupts.
- Keyboard and screen interactions use the simple UNIX command line protocol, or else the ANSI.SYS driver is used for any complex screen operations.
- The program always treats 32-bit pointers as atomic objects, i.e., it does not manipulate the segment and offset portions separately.

These same rules apply to assembly language functions called by C programs, although it's usually more difficult to tell if such functions are well-behaved by just "eyeballing" them. So, in addition to the items listed above, you should also look out for these coding practices, which will complicate the conversion:

- INT instructions, which generate software interrupts;
- IN and OUT instructions, which directly access the I/O ports;
- CLI and STI instructions, which change the state of the interrupt enable flag;
- Instruction sequences that change segment register contents, such as

```
MOV   AX,ES ; advance to next paragraph
INC   AX
MOV   ES,AX
```

  which are guaranteed to fail because segment register arithmetic is different in protected mode; and
- Timing loops, such as

```
          MOV    CX,1000   ; wait awhile
DELAY:    LOOP   DELAY
```

  which are risky, even under DOS because they become invalid when you upgrade to a faster computer.

The first three of these five assembly-language coding practices will cause protection violations because OS/2 does not allow normal processes to execute INT, IN, OUT, CLI, and STI instructions. The fourth will usually cause addressing exceptions because the process will attempt to load an invalid selector into a segment register. The fifth just causes the program to behave erratically, since the time delay varies depending on how heavily OS/2 is loaded.

We discovered that much of our assembly-language code had none of these problems and was written at that level just to achieve the maximum speed and minimum size. The Lattice floating point library is an example; it moved to OS/2 with no changes at all.

If, however, you find any of these transgressions, you'll have to do some redesigning. Fortunately, such functions are usually simple, and you can often rewrite them in C, making direct calls to the OS/2 Application Program Interface (API). That is, many of these errant assembly-language routines were originally written just to get at the DOS API in some fashion not otherwise supported by the standard C libraries. But under OS/2, you can usually find an equivalent API function and call it directly from C.

So, if you're lucky enough to have only well-behaved programs, with just a few assembly-language changes to worry about, stop reading here and start converting. Of course, well-behaved programs are typically command-line oriented, and are often somewhat boring to use. So, you'll pay your OS/2 dues later if you decide to spiff these programs up by adapting them to the Presentation Manager—a not-so-trivial task.

2) Highly Interactive Programs

The DOS world is full of programs that do all sorts of magic with the keyboard, screen, and mouse. In fact, one of the things that has made DOS so popular is the ease with which such software can be written. Of course, this has less to do with DOS than it does with the *de facto* hardware standards caused by the IBM PC. The undeniable influence of IBM has simplified many of the aspects of interactive programming that were quite chaotic in the earlier CP/M world, such as memory-mapped video techniques, keyboard scan codes, and extended character sets.

Nonetheless, DOS programs that have a lot of user interaction generally present more OS/2 conversion difficulties than the well-behaved programs discussed above. This is because all interfaces with the BIOS and video RAM must be changed. However, if you've followed the recommended practice of isolating these into a few modules, you'll only be dealing with a small percentage of the computer code.

For example, the Lattice Unicalc Spreadsheet (about 25,000 C source lines) and the Lattice Screen Editor (about 15,000 C source lines) are both highly interactive, screen-oriented programs that write directly to the video buffers. Yet, each required less than one person-week to convert, mainly because the human interfaces were designed in a very modular way. Surprisingly, there was no perceivable loss of performance relative to the DOS versions, as long as OS/2 was not running other tasks.

To convert an interactive program, you must replace all BIOS interrupts with calls to the corresponding OS/2 keyboard (*Kbd*), mouse (*Mou*), and video (*Vio*) functions listed in Figures 1, 2, and 3. These interrupts are usually generated directly by assembly-language functions, or indirectly from C programs via the *int86* function. Then you must modify all routines that directly address the video RAM.

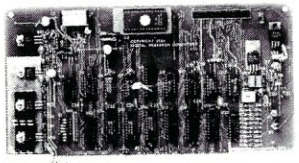
ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

S100 EPROM PROGRAMMER

OUR NEWEST DESIGN, FOR FAST EFFICIENT PROGRAMMING OF THE MOST POPULAR EPROM'S ON YOUR S100 MACHINE. COMES WITH MENU DRIVEN SOFTWARE THAT RUNS UNDER CP/M 2.2 (8 INCH). PC BOARD SET CONSISTS OF (S100) MAIN LOGIC BOARD REMOTE PROGRAMMING CARD AND SIX PERSONALITY MINI BOARDS FOR 2716, 2532, 2732, 2732A, 2764, AND 27128. SOLD AS BARE PC BOARD SET ONLY WITH FULL DOC. SOFTWARE FEATURES "FAST" PROGRAMMING ALGORITHM. FOR Z80 BASED SYSTEMS.



PC BOARD SET, FULL DOCUMENTATION, 8 IN. DISKETTE WITH SOFTWARE.

PRICE CUT! **$39.95**

128K S100 STATIC RAM/EPROM BOARD

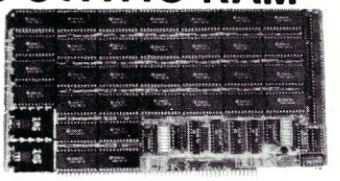
JUST OUT! USES POPULAR 8K X 8 STATIC RAMS (6264) OR 2764 EPROMS. FOR 8 OR 16 BIT DATA TRANSFERS! IEEE 696 STANDARD. LOW POWER. KITS ARE FULLY SOCKETED. FULL DOC AND SCHEMATICS INCLUDED. 24 BIT ADDRESSING.

NEW!

$59.95	$219.00	$139.00
BARE PC BOARD	128K RAM KIT	128 EPROM KIT

64K S100 STATIC RAM

$99.00 KIT



LOW POWER!

150 NS ADD $10

BLANK PC BOARD WITH DOCUMENTATION $49.95

SUPPORT ICs + CAPS $17.50

FULL SOCKET SET $14.50

FULLY SUPPORTS THE NEW IEEE 696 S100 STANDARD (AS PROPOSED)

ASSEMBLED AND TESTED ADD $50

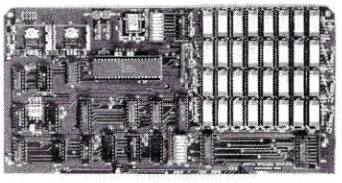
FEATURES: ***PRICE CUT!***

- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports IEEE 696 24 BIT Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- 2716 EPROMs may be installed in any of top 48K.
- Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- BOARD may be partially populated as 56K.

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



BLANK PCB (WITH CP/M* 2.2 PATCHES AND INSTALL PROGRAM ON DISKETTE) $24.95 (8203-1 INTEL $29.95)

FEATURES:

- 256K on board, using +5V 64K DRAMS.
- Uses new Intel 8203-1 LSI Memory Controller.
- Requires only 4 Dip Switch Selectable I/O Ports.
- Runs on 8080 or Z80 S100 machines.
- Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
- Provisions for Battery back-up.
- Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
- The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- Compare our price! You could pay up to 3 times as much for similar boards.

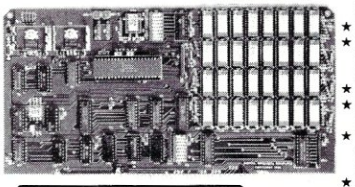
CLOSE OUT! BLANK PCB ONLY: **$24.95**

#LS-100

1 MEG. S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

LS 100 II ***NEW!***



BLANK PCB (WITH CP/M* 2.2 PATCHES AND INSTALL PROGRAM ON DISKETTE) $59.95 (8203 1 INTEL $29.95)

FEATURES:

- 1 Meg. on board, using +5V 256K DRAMS. (With Parity)
- Uses new Intel 8203-1 LSI Memory Controller.
- Requires only 4 Dip Switch Selectable I/O Ports.
- Runs on 8080 or Z80 S100 machines.
- Up to 4 LS-100 boards can be run together for 4 Megs. of On Line Solid State Disk Storage.
- Provisions for Battery back-up.
- Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
- The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- Compare our price! You could pay up to 3 times as much for similar boards.

(ADD $50 FOR A&T) **$259.00**

#LS-100 II (FULL 1 M.B. KIT)

1 MEGA BYTE!

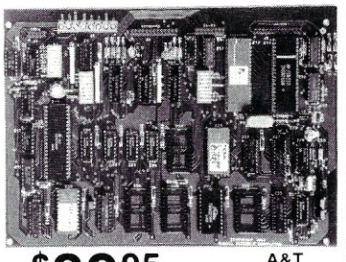
ZRT-80 CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- RS232 at 16 BAUD Rates from 75 to 19,200.
- 24 x 80 standard format (60 Hz).
- Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- Composite or Split Video.
- Any polarity of video or sync
- Inverse Video Capability.
- Small Size: 6.5 x 9 inches.
- Upper & lower case with descenders.
- 7 x 9 Character Matrix.
- Requires Par. ASCII keyboard.

FOR 8 IN. SOURCE DISK OR PC-XT FORMAT 5¼ IN. ADD $10



$89.95 #ZRT-80 (COMPLETE KIT, 2K VIDEO RAM)

A&T ADD $50

OUR BEST SELLER!

THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

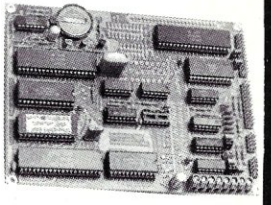
★ FROM LINGER ENTERPRISES ★

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. This highly versatile board can be used as a stand alone video terminal, or without a keyboard, as a video console. VT100, VT52 Compatible.

FEATURES:

- Uses the new CRT9128 Video Controller driven by a 6502A CPU
- On-Screen Non-Volatile Configuration
- 10 Terminal Modes: ANSI, H19, ADM-5, WYSE 50, TVI-920, KT-7, HAZ-1500, ADDS 60, QUME-101, and Datapoint 8200
- Supports IBM PC/XT, and Parallel ASCII Keyboards
- Supports standard 15.75 kHz (Horiz.)
- Composite or Split Video (50/60 Hz)
- 25 X 80 Format with Non-Scrolling User Row
- Jump or Smooth Scroll
- RS-232 at 16 Baud Rates from 50 to 19,200
- On Board Printer Port
- Wide and Thin Line Graphics
- Normal and Reverse Screen Attributes
- Cumulative Character Attributes: De-Inten, Reverse, Underline and Blank
- 10 Programmable Function Keys and Answerback message
- 5 X 8 Character Matrix or 7 X 9 for IBM Monitors
- Mini Size: 6.5 X 5 inches
- Low Power: 5VDC @ .7A, ± 12VDC @ 20mA.

MICRO SIZE!



$79.95 FULL KIT

w/100 Page Manual

ADD $40 FOR A&T

OPTIONAL EPROM FOR PC/XT STYLE SERIAL KEYBOARD: $15

SOURCE DISKETTE: PC/XT FORMAT 5¼ IN. $15

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

TERMS: Add $3.00 postage. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 6-1/4% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50 add 85¢ for insurance.

*TM OF DIGITAL RESEARCH INC. (CALIF.)

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH INC. (CALIF.) THE SUPPLIERS OF CPM SOFTWARE

2.1) Converting The Keyboard Interface

Adapting to the OS/2 keyboard interface is usually fairly straightforward and has no noticeable effect on performance. Listing 1 is a piece of DOS-based C code that obtains a character and scan code via BIOS interrupt 16H, using *int86:*. The same operation in OS/2 can be coded as in Listing 2.

Although the OS/2 version looks more complicated, it actually is not. Most of the OS/2 statements are declarations that would normally be coded once and kept in header files. In fact, the OS/2 Programmer Toolkit includes such header files for both C and assembly language.

The call to *KbdCharIn* probably seems mysterious if you are unfamiliar with the OS/2 Application Program Interface (API). I won't cover the API in detail here, but a few words of explanation should enable you to understand the remaining examples.

First, note that the function is declared with the "far" and "pascal" keywords. The former tells the compiler to use a "far call," while the latter specifies the so-called "pascal calling sequence." This means that arguments are pushed on the stack from left to right and that the function pops the arguments before it returns.

KbdCharIn takes three arguments. The first argument is a 32-bit pointer to the structure in which OS/2 will return information about the next keystroke. The OS/2 API requires that all addresses be passed in full 32-bit form, and so you must make sure that your C

Figure 1. OS/2 Keyboard Functions

FAM	BIOS INT 16H	NAME	DESCRIPTION
F	00H	KbdCharIn	Read character and scan code
–	–	KbdClose	Close logical keyboard
–	–	KbdDeRegister	De-register keyboard subsystem
–	–	KbdFreeFocus	Free keyboard focus
F	–	KbdFlushBuffer	Flush keystroke buffer
–	–	KbdGetFocus	Get keyboard focus
F	01H	KbdGetStatus	Get keyboard status
–	–	KbdGetCP	Get keyboard code page ID
–	–	KbdOpen	Open logical keyboard
R	02H	KbdPeek	Peek at character and scan code
–	–	KbdRegister	Register keyboard subsystem
–	–	KbdSetCP	Set keyboard code page ID
–	–	KbdSetCustXt	Set custom translate table
–	–	KbdSetFgnd	Set foreground keyboard priority
F	–	KbdSetStatus	Set keyboard status
–	–	KbdShellInit	Initialize shell
F	–	KbdStringIn	Read character string
–	–	KbdSynch	Synchronize keyboard access
–	–	KbdXlate	Translate scan code

Figure 2. OS/2 Mouse Functions

INT 33H	NAME	DESCRIPTION
–	MouClose	Close mouse device
–	MouDeRegister	Deregister a mouse subsystem
01H	MouDrawPtr	Draw a pointer
–	MouFlushQue	Flush mouse event queue
–	MouGetDevStatus	Get mouse device status flags
–	MouGetEventMask	Get mouse event mask
–	MouGetHotKey	Get mouse hot key definition
–	MouGetNumButtons	Get number of mouse buttons
–	MouGetNumMickeys	Get number of mickeys per centimeter
–	MouGetNumQueEl	Get number of mouse event queue elements
03H	MouGetPtrPos	Get mouse pointer position
–	MouGetPtrShape	Get mouse pointer shape
–	MouGetScaleFact	Get mouse scaling factors
–	MouInitReal	Initialize real-mode mouse driver
00H	MouOpen	Open mouse device
05H,06H	MouReadEventQue	Read mouse event queue
–	MouRegister	Register a mouse subsystem
02H	MouRemovePtr	Remove mouse pointer from a screen area
–	MouSetDevStatus	Set mouse device status flags
0CH	MouSetEventMask	Set mouse event mask
–	MouSetHotKey	Set mouse hot key definition
04H	MouSetPtrPos	Set mouse pointer position
09H,0AH	MouSetPtrShape	Set mouse pointer shape
–	MouSetScaleFact	Set mouse scaling factors
–	MouShellInit	Initialize shell linkage
–	MouSynch	Synchronize mouse subsystem

Figure 3. OS/2 Video Functions

FAM	BIOS INT 10H	NAME	DESCRIPTION
–	–	VioDeRegister	De-register video subsystem
–	–	VioEndPopUp	De-allocate pop–up display screen
–	–	VioGetANSI	Get ANSI status
F	–	VioGetBuf	Get logical video buffer
–	–	VioGetConfig	Get video configuration
–	–	VioGetCp	Get video code page ID
F	03H	VioGetCurPos	Get cursor position
F	–	VioGetCurType	Get cursor type
–	–	VioGetFont	Get font
F	0FH	VioGetMode	Get display mode
F	–	VioGetPhysBuf	Get physical display buffer
–	–	VioGetState	Get video state
–	–	VioModeUndo	Cancel mode wait
–	–	VioModeWait	Wait for mode change
–	–	VioPopUp	Allocate pop-up display screen
–	–	VioPrtSc	Print screen
–	–	VioPrtScToggle	Print screen key operation trap
F	08H	VioReadCellStr	Read cell string
F	–	VioReadCharStr	Read character string
–	–	VioRegister	Register video subsystem
–	–	VioSavRedrawUndo	Cancel save-redraw wait
–	–	VioSavRedrawWait	Wait for save-redraw
–	–	VioScrLock	Lock the screen
F	07H	VioScrollDn	Scroll down
F	–	VioScrollLf	Scroll left
F	–	VioScrollRt	Scroll right
F	06H	VioScrollUp	Scroll up
–	–	VioScrUnLock	Unlock the screen
–	–	VioSetANSI	Set ANSI mode on or off
–	–	VioSetCp	Set video code page ID
F	02H	VioSetCurPos	Set cursor position
F	01H	VioSetCurType	Set cursor type
–	–	VioSetFont	Set font
F	00H	VioSetMode	Set display mode
–	0BH	VioSetState	Set video state
F	–	VioShowBuf	Display logical buffer
F	09H	VioWrtCellStr	Write cell string
F	0AH	VioWrtCharStr	Write character string
F	–	VioWrtCharStrAttr	Write character string with attribute
F	–	VioWrtNAttr	Write N attributes
F	–	VioWrtNCell	Write N cells
F	–	VioWrtNChar	Write N characters
F	0EH	VioWrtTTY	Write a TTY string

F = Full family-mode support, R = Restricted family-mode support

code expands 16-bit pointers to that 32-bit form. The cast operation

```
(char far *)(&key)
```

does exactly that to the address of the key structure.

The second argument tells *KbdCharIn* what to do if no keystroke is ready. A value of 0 causes the function to wait until the user presses a key.

The third argument is the "logical keyboard handle," which should be 0 for the default keyboard. When OS/2 is managing several applications (called "sessions") at the same time, it gives each one its own keyboard buffer, called "logical keyboard 0," which will receive input characters when that application is in the foreground, i.e., is visible to the user. In some cases an application can request additional logical keyboards, which will have handles greater than 0.

2.2) Converting The Mouse Interface

If you access the mouse under DOS via software interrupt 33H, the OS/2 conversion is very similar to the keyboard work described above. Figure 3 shows which *Mou* calls in OS/2 correspond to the interrupt 33H operations.

However, if you rely upon a special mouse interface provided by a DOS driver or pseudo-driver, you're in for some redesign. Most DOS mice on the market today include the necessary software to handle the interrupt 33H protocols, and I'd suggest that you change your DOS code to use that approach. Then it should be a straightforward job to convert that code to OS/2 using the *Mou* functions.

2.3) Converting Video BIOS Calls

If your DOS program accesses the video subsystem through BIOS interrupts, then the conversion to OS/2 is, once again, very similar to the keyboard work described above. For example, Listing 3 is a DOS code snippet that clears the screen and moves the cursor to the upper left corner. The same operation in OS/2 can be accomplished by the sequence shown in Listing 4.

VioScrollUp accepts seven arguments. The first four define the scrolling area, that is, the upper left and lower right corners of the block being scrolled. Here we're dealing with the entire screen of 25 rows and 80 columns, and so the upper left and lower right corners are (0,0) and (24,79), respectively. The fifth argument specifies the number of lines to scroll, 25 in this case. The sixth argument is the attribute and character used to fill the new lines on the screen. The attribute value here is 7 for white-on-black, and the character value is 0x20, which is a blank.

The final argument to *VioScrollUp* and most other *Vio* calls is the "logical screen handle." As with the keyboard, the OS/2 session manager provides a separate video buffer for each active session. The handle for this default buffer is 0. The application program can create additional video buffers if necessary, and these will have handles greater than 0.

VioSetCurPos is much simpler. It positions the cursor to the row and column specified by the first two arguments. The third argument is the logical screen handle.

2.4) Converting Video RAM Accesses

If your program paints the screen by writing directly into the video RAM, you must deal with the fact that OS/2 severely restricts your access to this special area of memory. When you run in protected mode, the video RAM doesn't even appear in your memory map, and so if you try something like writing into the monochrome video buffer at *segment:offset* B000:0000, OS/2 will abort your process with an addressing exception.

Listing 5 is an example of some non-portable DOS C code for clearing the monochrome screen. The strange-looking "union" is used to form a valid *segment:offset* pointer to the video buffer. Then the loop simply resets that area to blanks with white-on-black attributes.

Now, if you try to run this code in OS/2 protected mode, you'll see a prime example of an OS/2 abort message. But text-mode programs like this one can be easily converted to OS/2; you just write into the logical video buffer and let OS/2 update the physical buffer. This technique is really nothing new; it's exactly what you have to do in DOS-based multitasking systems such as IBM Topview and Microsoft Windows. For example, the code in Listing 6 offers a Topview-compatible example. Here we use video BIOS function 254, which leaves ES:DI intact if Topview or some other session management system is not installed. Otherwise, ES:DI is loaded with the *segment:offset* pointer to the logical video buffer. Then, after the buffer is cleared, we use video BIOS function

Listing 1. DOS based C code that obtains a character and scan code via BIOS interrupt 16H.

```
union REGS r;           /* define register set */

r.h.ah = 0:             /* use code 0 in AH to get next
                           character */
int86(0x16,&r,&r);      /* generate INT 16H */
scancode = r.h.ah;      /* save scan code from AH */
charcode = r.h.al;      /* save character code from AL */
```

Listing 2. OS/2 based C code that obtains a character and scan code via BIOS interrupt 16H.

```
extern far pascal KbdCharIn();

struct CHARDATA         /* structure to hold data from
                           KbdCharIn */

  char acode;           /* ASCII character code */
  char scode;           /* scan code */
  char status;          /* device status */
  char nls;             /* shift state for National
                           Language Support */
  short shift;          /* keyboard shift state */
  long time;            /* timestamp */
  key;

int error;              /* error code */

error = KbdCharIn((far char *)(&key),0,0);  /* call
                           OS/2 keyboard service */

if(error == 0)

  scancode = key.scode;      /* save scan code */
  charcode = key.acode;      /* save character code */

else ...                    /* error handler goes here*/
```

Listing 3. DOS code to clear the screen and return the cursor to the upper left corner.

```
union REGS r;         /* define register set */

r.x.ax = 0x0619;      /* AH=6 for scroll up, AL=25 for
                         entire screen */
r.h.bh = 7;           /* BH=7 for white-on-black
                         attribute */
r.x.cx = 0;           /* (CH,CL) is upper left
                         (row,col)*/
r.x.dx = 0x184F;      /* (DH,DL) is lower right
                         (row,col)*/
int86(16,&r,&r);      /* generate video interrupt */

r.h.ah = 2;           /* AH=2 to set cursor position */
r.h.bh = 0;           /* BH=0 for page 0 */
r.x.dx = 0;           /* (DH,DL) is (row,col) */
int86(16,&r,&r);      /* generate video interrupt */
```

Listing 4. OS/2 code to clear the screen and return the cursor to the upper left corner.

```
extern far pascal VioScrollUp();
extern far pascal VioSetCurPos();
int error;

error = VioScrollUp(0,0,24,79,25,0x0720,0);
if(error) ...

error = VioSetCurPos(0,0,0);
if(error) ...
```

Listing 5. Non-portable DOS C code for clearing the monochrome screen.

```
int i;
union

  unsigned short x[2];
  unsigned short far *p;
  q;

q.x[0] = 0;          /* form segment:offset pointer to
                        video RAM */
q.x.[1] = 0xb000;

for(i = 0; i < 2000; q.p[i++] = 0x0720);
```

Listing 6. Topview code to clear the screen.

```
int i;
union REGSS r;          /* define register set with
                           segments */
union                   /* used to make video RAM pointer */

  unsigned short x[2];
  unsigned short far *p;
  q;

r.x.di = 0;             /* set ES:DI to video buffer */
r.x.es = 0xb000;
r.h.ah = 0xfe;          /* use BIOS function 254 */
int86s(16,&r,&r);       /* ask BIOS for logical buffer
                           pointer */

q.x[0] = r.x.di;        /* form far pointer to logical
                           buffer */
q.x.[1] = r.x.es;

for(i = 0; i < 2000; q.p[i++] = 0x0720);  /* clear the
                                             buffer */

r.x.di = q.x[0];        /* set ES;DI to start of logical
                           buffer */
r.x.es = q.x[1];
r.x.cx = 4000;          /* set size to 25 rows, 80
                           columns */
r.h.ah = 255;           /* use BIOS function 255 */
int86s(16,&r,&r);       /* ask BIOS to update screen */
```

Listing 7. Topview code to clear the screen adapted for OS/2.

```
extern far pascal VioGetBuf();
extern far pascal VioShowBuf();

int i,error;
short far *p;
short size;

error =                       /* get logical video
                                 buffer ptr */
  VioGetBuf((long far *)(&p),(short far *)(&size),0);
if(error) ...

for(i = 0; i < (size/2); p[i++] = 0x0720); /* clear the
                                              buffer */

error = VioShowBuf(0,size,0);             /* update the
                                             screen */
if(error) ...
```

Listing 8. A program to directly access the physical video buffer.

```
extern far pascal VioGetPhysBuf();
extern far pascal VioScrLock();
extern far pascal VioScrUnlock();

int i,error;
struct physdata
```

... continued on next page

255 to update the screen. If no session manager is active, function 255 does nothing, since we've actually been writing to the physical buffer.

The equivalent OS/2 code (Listing 7) looks quite similar, except that the BIOS interrupts are replaced with *Vio* calls. *VioGetBuf* returns a pointer to the logical video buffer. The pointer is stored into the 32-bit area whose 32-bit address is given by the first argument. That's the reason for the *(long far *)* cast in that argument position. The second argument is a 32-bit pointer to a short integer where the buffer length (in bytes) will be returned, and again, a cast operation is needed. The logical video handle completes the argument list, as we've already discussed.

After clearing the logical buffer, you must use *VioShowBuf* to get the physical screen updated, which will actually occur only if your program is running as the foreground session. The first argument is the offset into the logical buffer, and the second is the number of bytes being updated.

So, if you were prudent enough to incorporate this type of Topview logic into your screen-oriented programs, you'll have an easy time moving them to OS/2. However, if you scattered absolute addresses of video buffers throughout your code, you'll have to do some clean up first. I'm ashamed to admit that we had a few programs like that, and we found it worthwhile to first clean them up under DOS using the techniques already described. After that, the move to OS/2 was painless, and we improved our DOS versions as well.

2.5) Converting Graphic Programs

So far, we've only discussed text-oriented programs, that is, those programs that use the non-graphical video display modes. While OS/2 Standard Edition 1.0 provides excellent support for such applications, it has little to offer if your programs generate graphical displays.

However, the situation is not completely hopeless. Contrary to what some would have you believe, you do not have to wait for the Presentation Manager before you can move graphical programs to OS/2. The new operating system does provide the means to gain direct access to the physical video buffer and the display adaptor's I/O ports, both of which are necessary for generating graphical displays. This is not too surprising, since the Presentation Manager is itself an application on top of the OS/2 base, and it needs these capabilities.

A detailed discussion of low-level graphics programming under OS/2 would be too lengthy for this article, but Listing 8 offers a small example of how you can directly access the physical video buffer. *VioGetPhysBuf* maps the specified video buffer into your address space, if possible, and returns one or more selectors that you can use to form 32-bit pointers for protected-mode access. One selector is returned for each 64 kilobytes of video buffer area. The mapping, however, is only valid when your program is executing in the foreground, that is, when it has control of the actual display screen. So, you must call *VioScrLock* before accessing the buffer and

```
... continued
q.x[0] = 0;                          /* form far pointer to
                                        buffer */
q.x[1] = x.prots[0];

error = VioScrLock(1,(char far *)(&i),0);
                                     /* lock the screen */
if(error) ...

for(i = 0; i < 2000; q.p[i++] = 0x0720);
                                     /* clear the buffer */

error = VioScrUnlock(0);
                                     /* unlock the screen */
if(error) ...
  unsigned long real;                /* physical video
                                        buffer address */
  unsigned long length;              /* buffer length */
  unsigned short prots[2];           /* protected-mode
                                        buffer selectors */
  x;

union                                /* used to make video
                                        RAM pointer */

  unsigned short x[2];
  unsigned short far *p;
  q;


x.real = 0xb0000;                    /* load physical
                                        address of buffer */
x.length = 4000;                     /* load buffer length*/
error = VioGetPhysBuf((struct physdata far *)(&x),0);
                                    /* Map buffer */
if(error) ...
```

VioScrUnlock when you are finished. The former, as used in this example, does not return until the user brings the program to the foreground via the session manager. Then it locks the screen so that the user cannot switch to another session until the current program unlocks the screen via *VioScrUnlock*.

Now, even though the example in Listing 8 writes to the physical video buffer in text mode, you can use this same technique to draw graphic images. I won't present an example here because there are so many subtle points that this topic deserves an article of its own, however, the general approach is:

- Use *VioGetConfig* to determine the type of display adaptor equipped. *VioGetMode* and *VioGetState* might also be needed to determine the current display mode and state.
- Use *VioSetMode* and, if necessary, *VioSetState* to establish the appropriate graphical mode.
- Create separate execution threads using *VioModeWait* and *VioSavRedrawWait* to re-establish the display mode and save or restore the display data when a session switch occurs.
- Use *VioGetPhysBuf*, *VioScrLock*, and *VioScrUnlock* as above to gain access to the physical video buffer in order to draw the graphic images.

If you need to access the display adaptor's I/O registers, as is often the case with EGA displays, then you'll also have to create a function with I/O privileges that you can then call from your program. This is necessary because normal OS/2 processes are not allowed to execute I/O instructions.

As I said, graphic programming can be done in OS/2 without relying upon the Presentation Manager. It isn't as easy as displaying text alone, but if you study the OS/2 Technical Reference thoroughly and try a few simple experiments, you'll quickly get the hang of it.

3) Terminate-Stay-Resident (TSR) Programs

TSR programs are a special category of highly interactive DOS applications. The typical TSR, such as Borland SideKick or Lotus Metro, is dormant until the user activates it using a hot key. Then it opens a window on the screen and enters into a dialogue with the user to accomplish some task, such as updating an appointment calendar. When this dialogue is finished, the TSR restores the original screen contents and goes back to sleep. So, TSR programs typically have the same OS/2 conversion problems as the interactive programs discussed earlier; that is, their keyboard, mouse, and screen access techniques must be changed to meet the OS/2 requirements.

In addition, you need to decide if you want the TSR to become a normal OS/2 process or a "monitor" process. If the TSR is providing a data management service, such as an appointment calendar, you could simply load it as an

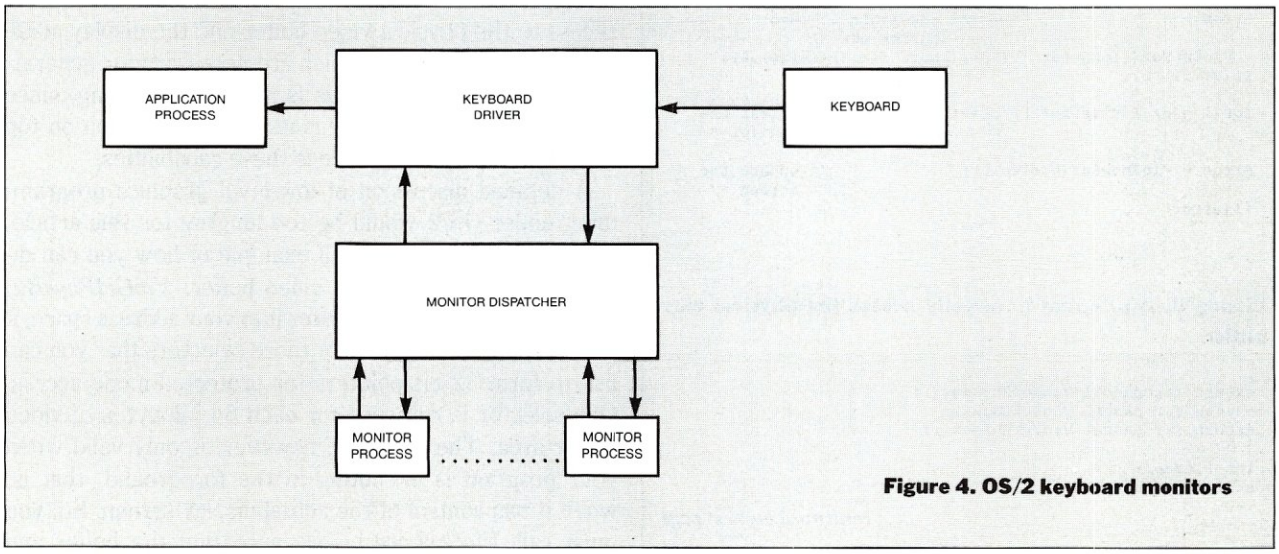


Figure 4. OS/2 keyboard monitors

OS/2 session and use the session manager hot keys to activate the calendar.

However, if the TSR has to continuously monitor the keyboard input, as a keyboard macro package such as Prokey must do, then you need to hook it into the keyboard I/O driver as a monitor process. Fortunately, OS/2 provides an elegant way to do this which eliminates the TSR contention problems that often occur under DOS. Figure 4 shows how this works.

The monitor process calls *DosMonOpen* and *DosMonReg* to open a monitor chain in the device driver and register itself as a monitor. Then each time the device driver receives a keystroke, it sends a "keystroke data packet" out to the first monitor in the chain. That monitor reads the packet via *DosMonRead* and chooses to absorb the keystroke, pass it on, or substitute one or more pseudo-keystrokes. These latter two operations are done via *DosMonWrite*, which sends the keystroke packet on to the next monitor in the chain. Unless everything gets absorbed in the chain, the original or substitute keystrokes eventually reach the end of the chain and are returned to the device driver, which passes them on to the application program that is waiting for keyboard data.

As with graphical techniques discussed above, there are many subtleties you must keep in mind when designing a monitor process, and so I won't attempt a thorough description here. The main thing to keep in mind, though, is that a monitor must be very fast or else the system will appear to be sluggish. Also, note that you can hook monitors into some of the other standard character drivers. For example, the OS/2 print spooler is implemented as a monitor process hooked into the printer driver.

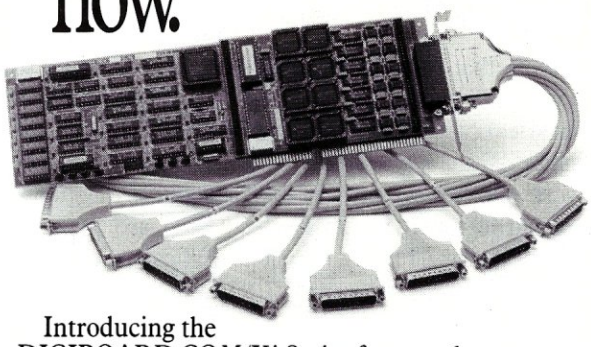
4) I/O Drivers

I/O drivers present the toughest conversion challenges, if only because they hook into the messy guts of DOS and typically require a large amount of assembly language programming. Furthermore, as with DOS TSR programs, the rules for creating drivers have not been well documented, and so designers have had to use *ad hoc* techniques learned by experimentation.

Unfortunately, OS/2 drivers tend to be much more complicated than their DOS counterparts because they must operate in a multitasking environment supporting both real-mode and protected-mode processes. Because of these significant environmental differences, the typical DOS driver doesn't survive the conversion, except for small sequences of code that deal directly with the device. In other words, if you write drivers for a living, roll up your sleeves because you've got a lot of programming work ahead of you.

We were initially concerned that the techniques for creating OS/2 drivers would remain mysterious and that we could not obtain the knowledge experimentally because OS/2 is so well-protected. However, IBM's documentation was a pleasant surprise, because it explains this arcane topic in great detail and includes a fairly complete example. Microsoft has also been providing a driver

Introducing multi-channel communications boards 400% faster than what you're probably using now.



Introducing the DIGIBOARD COM/Xi Series front-end processor. Intelligent multi-channel communications boards 400% faster than the industry standard.

Like our popular COM/X Series, they provide users of PC/XT/AT-compatible computers with four or eight individually addressable serial ports. But with the new COM/Xi series we've added:

- an 80188 co-processor operating at 10 MHz
- 256K of dual-ported RAM + 16K of ROM, all accessible to user/ programmers for application and security software development
- a modular design that allows us to custom-tailor I/O to individual customer requirements.

On-board intelligence means more speed for multi-user operating systems and multi-channel data collection and dissemination.

And makes the new DIGIBOARD COM/Xi Series a more intelligent choice for you.

DigiBoard
Plugging you into Tomorrow.

Call **1-800-344-4273.** In Minnesota, (612) 922-8055.

development kit, including an OS/2 kernel debugger, to attendees of its driver design course.

Family Mode Considerations

The final topic you must consider when converting a DOS program to OS/2 is whether you want the result to be a "family-mode application." This means that the program will be put together in such a way that its executable, i.e., the .EXE file, will run correctly in OS/2 protected mode, OS/2 real mode, or under DOS 3.x. The OS/2 Programmer Toolkit includes the BIND utility program and a special library, API.LIB, to construct family-mode applications.

There are three simple rules for achieving family compatibility:

- Don't use any OS/2 API calls that are not part of the family-mode repertoire unless your program first tests that it is running in protected mode.
- Don't do any DOS-specific things, such as generate software interrupts, unless you verify that the program is running in real mode.
- Don't do any internal manipulation of 32-bit pointers unless you first check whether your program is in real or protected mode and choose the appropriate algorithm.

Note that the first rule requires that you know which functions in the C compiler library (and possibly other libraries) call API functions that are not allowed in family mode. When we converted the Lattice C Compiler, we were able to make most of the mainstream functions fully compatible with family mode. These are kept in the LC.LIB library. Lattice functions that can be used only in protected mode are kept in LCP.LIB, while those that can be used only in real mode are in LCR.LIB.

Figures 1, 2, and 3 indicate which of the keyboard, mouse, and video API functions are available in family mode. In the FAM column, an F means that the function is fully supported in the family mode, while an R indicates that there are some restrictions on family-mode use of the function. The OS/2 Technical Reference provides this information for all of the API functions.

Summary

Our experiences at Lattice in converting DOS applications to OS/2 protected mode have been much happier than we originally expected. While this exciting new operating system was designed with an eye to the future, it remains very compatible with the past. Good luck in your move to OS/2! □

David A. Schmitt spent 18 years at Bell Laboratories, where he was involved in the design of many real-time control systems, including a fault-tolerant version of UNIX. Since 1983 he has been President of Lattice, Incorporated, which developed the popular Lattice C Compiler and is now a subsidiary of SAS Institute.

Continued from page 23

link packages. Only one copy of any executable file, program, or library is loaded, no matter how many processes may be using it concurrently.

How a Process Finds the Handles

When a program is loaded, it is converted into a process with an address space that includes its loaded segments. The process is animated with a single thread that begins execution at the defined program entry point. The process finds out about the system using system calls that return information or give access to read-only data segments. One such, the environment segment, contains two blocks of strings: one that defines variables in the usual "name=value" form, and one that contains the program's executable file name and its command-line parameters.

The process receives three open file handles for standard input, standard output, and error reports (the standard printer file of MS-DOS is not supported by OS/2). These handles are prepared by the process's parent and will usually be associated with the screen group's keyboard and screen. If the parent is the command shell, the standard handles may have been opened to disk files or to pipes, buffered communication paths shared with other processes.

A process can set up its own handlers for the exception events in three classes. The first class is software signals generated by other processes, a class that includes the Control-C signal. The second class of events covers certain hardware traps. The third class is the Exit list, a list of procedures that will be called when a process terminates, regardless of the cause of termination. There are system calls to add a routine to the exit list, and to remove it, so a linked subsystem could install an exit handler while working for a process, and remove it when the process is finished.

An OS/2 system is host to a number of independent processes, each representing a unique address space. The processes, in turn, are home to a flock of threads that execute concurrently and asynchronously. A wealth of methods are available by which the processes may share and exchange data, and by which the threads may coordinate their activities.

OS/2 has been designed from the ground up to make it easy to add new subsystems. A new functional subsystem can be added in the form of a new dynamic linking library, and it becomes immediately accessible to every other part of the system. Such power and flexibility present a new programming challenge to systems integrators. But with perseverance and a little guidance, it will become as easy to find your way in OS/2 as it is to traverse the DOS landscape. □

This overview was condensed from The Programmer's Essential OS/2 Handbook *by David Cortesi, which is scheduled for publication this spring by M&T Books.*

THE PROGRAMMER'S SHOP

helps save time, money, and cut frustrations. Compare, evaluate, and find products.

FREE *Programmer's Update* Magazine

"Keeping Professional Programmers Current with Developer's Technology." *Programmer's Update* lives up to its motto witha viewpoint unique to this industry. We use our experience and contacts to bring you articles about intriguing software trends and technical issues, interviews with authors and innovators, news about products, surveys, insightful commentary and predictions, valuable resource listings. You can get a FREE sample copy today by calling our toll-free number. Mention "MS388". A personal subscription is just $25 a year.

Our Services:

- International Sales Desk
- Compare Products
- Help find a Publisher
- Evaluation Literature FREE
- Programmer's Update
- Dealers Inquire
- Newsletter
- Rush Order
- Over 700 products
- National Accounts Center

386 Development Tools

386 Assembler/Linker	PC	$ 389
386 Debug - by Phar Lap	PC	$ 129
386/DOS Extender	PC	$ 919
DESQview PS/2	PC	$ 109
F77L-EM - by Lahey	MS	Call
High C - by MetaWare	PC	Call
OS/286 & 386 by AI Architects	PC	Call
Paradox 386	MS	$ 799

AI Languages

APT - **A**ctive **P**rolog **T**utor - build applications interactively	PC	$ 49
ARITY Prolog - full, 4 Meg		
Interpreter - debug, C, ASM	PC	$ 229
COMPILER/Interpreter-EXE	PC	$ 569
Cogent Prolog Compiler	MS	$ 179
MicroProlog Prof. Comp./Interp.	MS	$ 609
PC Scheme LISP - by TI	PC	$ 85
Star Sapphire	MS	$ 429
TransLISP - learn fast	MS	$ 79
TransLISP PLUS	MS	$ 149
TURBO PROLOG by Borland	PC	$ 69

Others: IQ LISP ($239), IQC LISP ($269)

Basic

BAS_C - economy	MS	$ 179
BAS_PAS - economy	MS	$ 135
Basic Development Tools	PC	$ 89
db/Lib	MS	$ 119
Exim Toolkit - full	PC	$ 45
Finally - by Komputerwerks	PC	$ 85
Inside Track	PC	$ 49
Mach 2 by MicroHelp	PC	$ 55
NetWorks by Exim	PC	$ 89
QBase - screens	MS	$ 79
QuickBASIC	PC	Call
Quick Pak-by Crescent Software	PC	$ 59
Quick-Tools by BC Associates	PC	$ 109
Stay-Res	PC	$ 59
True Basic	PC	$ 79
Turbo BASIC - by Borland	PC	$ 69
Turbo BASIC Database Toolbox	MS	$ 69

FEATURES

SofTRAN, the Translation and Text Language by TransOptima - full procedural language like C plus pattern and nonprocedural constructs cuts development effort by up to 16 times. PC $ 349

FORCE III, Dbase Compiler by Sophco- small .EXEs, user-defined functions, I/O directives through BIOS/DOS/ANSI/ FORCE/user-defined, extensions include FOR..NEXT loops, soundex, 1D arrays. Maverick. PC $ 109

Program Objectively — for Less

Object-oriented programming tools free you from trivia tracking, make sophisticated applications practical quickly.

Consider **ACTOR**, full language and environment that lets you replace 100's of lines of code with just a few. It just may be the next standard language. Regularly $419 ($495 retail), **until March 31, 1988 . . . . . only $399.**

HOOPS by Ithaca Software lets you assign attributes to objects for advanced graphics applications. Include hidden line/surface removal, multiple light source rendering, and more in 3D with animation. Use natural language. Device independent, call from C, Pascal, or FORTRAN. Regularly $549 ($575 retail), **until March 31, 1988 . . . . . only $419.**

CALL TODAY for free detailed literature about these and other object-oriented tools. Take advantage of these special prices — **order before March 31, 1988.** Mention "MS388."

RECENT DISCOVERY

CLARION DBMS by Barrington Systems. Fast applications prototyping and development. Language, compiler, screen/report generators, editor, call other languages, read/write dBASE III + files. PC, List:$695

C Language-Compilers

AZTEC C86 - Commercial	PC	$499
C86 PLUS - by CI	MS	$359
Datalight Optimum - C	MS	$ 99
Instant-C/16M	PC	Call
Lattice C - from Lattice	MS	$259
Microsoft C 5.0- Codeview	MS	Call
Microsoft Quick C	MS	Call
Rex - C/86 standalone ROM	MS	$695
Turbo C by Borland	PC	$ 67

C Libraries-Files

BTree by Soft Focus	MS	$ 69
CBTREE - Source, no royalties	MS	$ 99
ctree by Faircom - no royalties	MS	$315
rtree - report generation	PC	$239
dB2C Toolkit V2.0	MS	$249
dbQUERY - ad hoc, SQL-based	MS	Call
dbVISTA - Object only	MS	Call
Source - Single user	MS	Call
dBx - translator	MS	$299

C-Screens, Windows, Graphics

C Worthy Interface Library	PC	$249
Curses by Aspen Scientific	PC	$109
dBASE Graphics for C	PC	$ 69
ESSENTIAL GRAPHICS - fast	PC	$185
FontWINDOW/PLUS	PC	$229
GraphiC - new color version	PC	$279
Greenleaf Data Windows	PC	$155
w/source	PC	$259
Terminal Mapping System	PC	$279
TurboWINDOW/C - for Turbo C	PC	$ 75
View Manager - by Blaise	PC	$199
Vitamin C - screen I/O	PC	$159
VC Screen	PC	$ 79
Windows for C - fast	PC	Call
Windows for Data - validation	PC	Call
ZView - screen generator	MS	$149

Atari ST & Amiga

We carry full lines of Manx & Lattice.

DBASE Language

Clipper compiler	PC	$399
dBase III Plus	PC	$429

Call for a catalog, literature, and *solid* value

800-421-8006

THE PROGRAMMER'S SHOP ™

Your complete source for software, services and answers

5-M Pond Park Road, Hingham, MA 02043
Mass: 800-442-8070 or 617-740-2510 1/88

RECENT DISCOVERY

CO/SESSION - Remotely access PC and peripherals, train or trouble-shoot from off-site, 2 users on one program. Session record/playback, file transfer, terminal emulation, keyboard and voice modes. PC $229

DBASE Language Cont.

dBASE III LANPack	PC	$649
DBXL Interpreter by Word Tech	PC	$ 99
FoxBASE + Dev. - V2.0	MS	$259
Quicksilver by Word Tech	PC	$369

DBASE Support

dAnalyst	PC	$ 89
dBase Tools for C	PC	$ 65
dBrief with Brief	PC	Call
dBC III by Lattice	MS	$169
dbug III	MS	$179
Documentor - dFlow superset	MS	$229
Genifer by Bytel-code generator	MS	$279
QuickCode III Plus	MS	$189
R&R Report Writer	MS	$139
Seek-It - Query-by-example	PC	$ 79
Silver Comm Library	MS	$139
Tom Rettig's Library	PC	$ 79
UI Programmer - user interfaces	PC	$249

DataBase & File Management

CQL	PC	$ 359
DataFlex by Data Access	PC	$ 899
DataFlex multiuser	PC	$1149
Magic PC	PC	$ 699
Paradox - original	PC	$ 369
Paradox V2.0	PC	$ 469
Revelation by Cosmos	PC	$ 779

Multilanguage Support

BTRIEVE ISAM	MS	$185
BTRIEVE/N-multiuser	MS	$455
GSS Graphics Dev't Toolkit	PC	$375
HALO Development Package	MS	$389
Graphics	PS	$209
Help/Control - on line help	PC	$ 99
HI-SCREEN XL	PC	$129
HOOPS Graphics Library	PC	$549
Informix 4GL-application builder	PC	Call
Informix SQL - ANSI standard	PC	Call
Instant Programmer's Help	MS	$ 79
NET-TOOLS - NET-BIOS	PC	$129
Opt Tech Sort - sort, merge	MS	$ 99
Norton Guides	PC	$ 75
Panel Plus	MS	$395
Pfinish - by Phoenix	MS	$209
Report Option - for Xtrieve	MS	$109
Screen Sculptor	PC	$ 89
SPSS/PC Plus	PC	$749
Synergy - create user interfaces	MS	$329
XQL - SQL for Btrieve	MS	$649
Xtrieve - organize database	MS	$179
ZAP Communications - VT 100	PC	$ 89

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. Formats: 3" laptop now available, plus 200 others. UPS surface shipping add $3/item.

Spooling on Novell Networks with Pascal

by Ed Rought and Tom Hoops

There are a number of advantages to providing network control over printers through spooler functions, such as those described in this article.

Providing solutions to multiple-user computer applications is often rewarding but difficult to accomplish. With users requiring true multiuser file access, multiple printers, internetwork message handling, and easy network management the new wave of personal computer-based networks provides real-world challenges to system integrators. As commercial application developers we have learned, often the hard way, that the solutions to these requirements fall to the people integrating the network and the users. To aid other system integrators in solving network applications, we will provide a case study of some of the problems we have encountered and solutions we have reached in print spooler control on Novell networks.

The System Requirements

In late 1986, after installing our office network using Novell Advanced Netware 2.0a, we were immediately faced with two projects that required multiple printer control. The first requirement was for our own office. We needed to provide printing capabilities for information requests received from our national product advertising. This task required that we print shipping labels at the workstation where the information request was entered, and that sales tracking records were printed immediately

The West Coast Computer Faire announces the first Computer Matchmaking Service.

You won't have to depend on fate at the 13th West Coast Computer Faire to find the products and services that are the perfect match for your needs.

We start you out on your path to high-tech bliss with Vertical Market Matching. We bring in the companies selling quality computers, software, peripherals, and add-ons—companies that meet the needs of people involved in specific business segments such as finance, medicine, manufacturing, law, education, engineering, and construction.

And our Product Matching makes it easy for you to find the software, add-ons and upgrades for the Commodore Amiga, Apple II or Macintosh, IBM PC/MS-DOS, IBM PS/2, Atari, Lotus and more, that will keep you happily gazing into your current system's eyes. Plus, we counsel you on the latest techniques and insights in our outstanding Conference sessions.

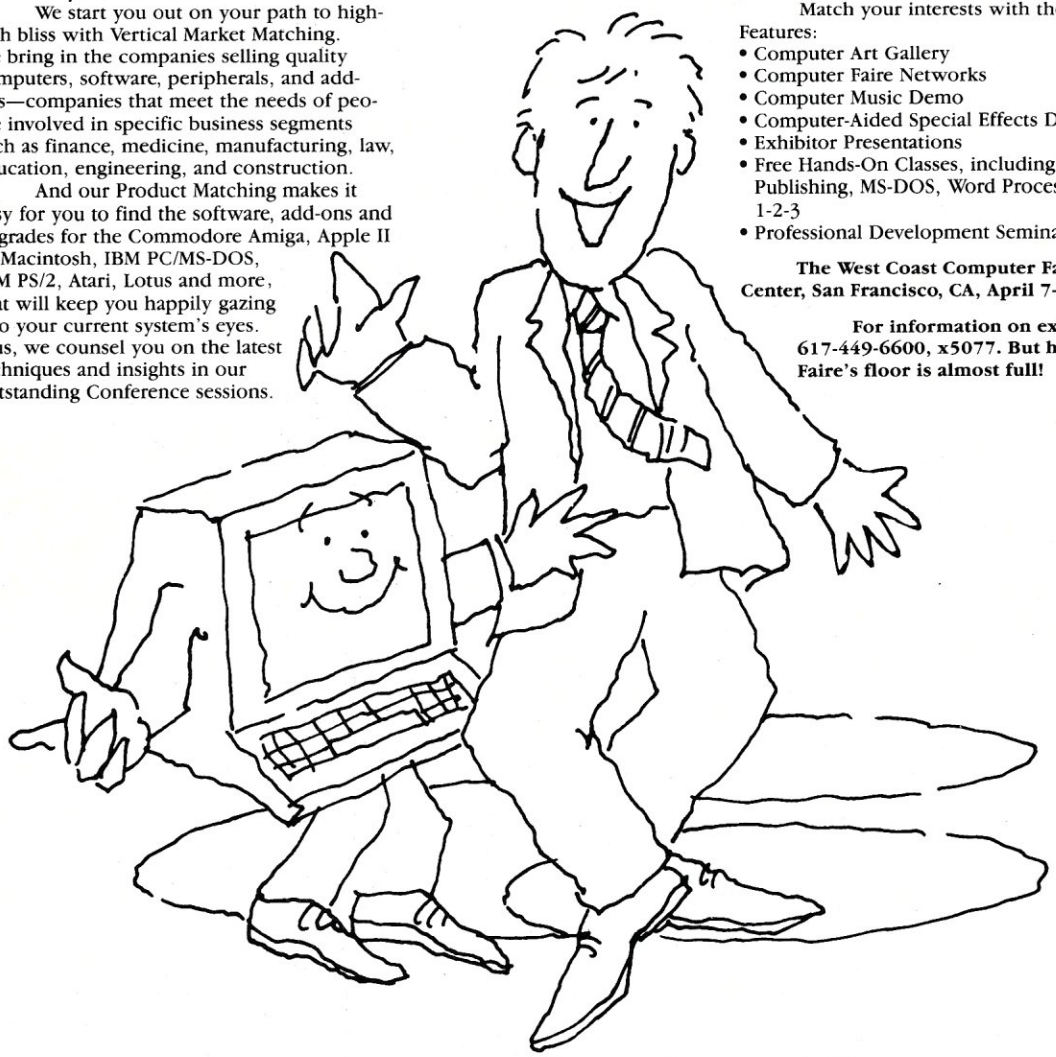
The West Coast Computer Faire has made and will make more matches than any another computer show. It's time we made the perfect match for you.

Match your interests with these Faire Features:

- Computer Art Gallery
- Computer Faire Networks
- Computer Music Demo
- Computer-Aided Special Effects Demo
- Exhibitor Presentations
- Free Hands-On Classes, including Desktop Publishing, MS-DOS, Word Processing, Lotus 1-2-3
- Professional Development Seminars

The West Coast Computer Faire, Moscone Center, San Francisco, CA, April 7-10, 1988

For information on exhibiting, call 617-449-6600, x5077. But hurry — the Faire's floor is almost full!



Register early and save $5!

Fill out this coupon and mail with your check(s), for $15.00 for each registrant, postmarked by March 17th, 1988. Include the names and addresses of registrants for whom you are enclosing a check. (Photocopy coupon for additional registrants.)

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Phone (_____) _____

Four day conference and exhibits $15.00 in advance. $20.00 at the door. Make check payable to "The West Coast Computer Faire." Mail to: Attendee Registration Department, The West Coast Computer Faire, 300 First Avenue, Needham, MA 02194. Advanced registrations accepted only with full payment and each registrant's name and address. Tickets will be mailed to each individual registrant separately.

THE 13TH WEST COAST COMPUTER FAIRE

April 7–10, 1988, Moscone Center, San Francisco, CA

after the labels. The second project involved creating an application for a local client that provided control of five printers by seven users on a network. The client would be using the workstations as point-of-sale terminals and the staff needed to print different forms based on the requirements of the customer being served at each station. Although the complexity of the two situations were different, the control solutions were very similar.

The Solution Options

Each case could be solved by a number of options. However, most of the potential solutions, other than network control, had major drawbacks. In our in-house prospect tracking system, we boiled the options down to three possible choices.

The first entailed providing multiple printers to each workstation with a manual- or software-controlled printer switch to control printer selection. This solution lowered software development costs by not requiring development of routines to interact with Netware. However, the hardware costs would be higher, since the network printers could not be utilized and a switch would be needed to control the printer selection. This option also raised the probability for operator error in selecting the proper printer for each form.

The second option for our tracking system was to provide one printer and have the operator manually change the paper before printing the material. This had the lowest cost because no additional hardware was required, and the software would be simple. However, the labor cost and lost time required to change paper far outweighed the savings on equipment and software.

The last and most practical solution was to use the Novell function calls to start and stop the spooler from within the application. This allows the local printing of labels when the spooler is disabled and the remote printing of sales tracking forms when the spooler is enabled. Otherwise, the Novell spooler will capture all printed outputs from the applications. The only disadvantage to using the spooler is the time required to create the new routines and test their functionality. The flexibility to split the task over the existing equipment, along with the potential for later adding additional printers to the file server to perform the task from any workstation, led us to select this option.

In our second project, the client with a point-of-sale problem required a similar kind of solution. The first possible option was to use a software-controlled printer switch to change printers when a command was received in the text stream. This option would lower software development costs, be transportable to a single user environment, and not be dependent on changes that might occur in network software. On the down side, the functionality would depend on another manufacturer, the hardware cost would be higher, control over the flow of print jobs would be poorer, there would be potential incompatibilities between switch manufacturers, and there would be a need for additional hardware. Of these negative side effects, relying on an additional manufacturer was our largest concern. In a market as volatile as the computer industry, we felt that entrusting our customer's functionality to a small group of printer switch manufacturers was not a reliable solution.

The next option we considered for our customer was to provide each workstation with its own set of printers. From the outset, this option was almost unthinkable, but we included it because it offered a viable solution if the software writers could not create the needed routines. Eventually, we discarded this solution because of the high cost and space requirements needed to provide three to five printers to each workstation.

We created routines to set spooler flags, obtain printer status, and modify and close the spooler device so we could have total control over the spooler functions.

The final option was to provide network control of each printer through the available network spooler functions. The advantages to using this approach include: transparent printer control, lower investment in hardware, more space for buffering of output documents, control by the system, and status checking for each print job.

Although this option sounds wonderful, there are disadvantages. First, extensive software development and an in-depth testing process are required. Secondly, there is the potential to create a backlog of print jobs as more workstations access the network printers. Third, the potential exists for a document, or portion of a document, to be lost if the Netware was not installed with sufficient communications buffers. The default for communication buffers was 40 packets or segments of transmitted information. We found that if seven users sent their documents at the same time, one to three of those documents would be lost. By increasing the buffers to 100 packets we were able to send all these documents safely. Since the buffers are allocated in the server's RAM, additional RAM may be required if the server is close to filling its present available RAM.

One Solution

We also opted for the network control approach because of the high cost of the other options. Our customer assisted us by determining the maximum number of users that would be accessing the five network printers at one time. We felt that installing a faster printer would elimi-

9-Track Tape Subsystem

The Data Interchange Solution You've Been Waiting For!

Qualstar's ½" 9-track Ministreamers™ bring full ANSI data interchange capability to your PC or Macintosh™ computer system.

With 9-track tape, you are free to exchange data files with any mainframe or minicomputer in the world.

Our affordable Ministreamers come in both 7" and 10½" versions and use less desk space than an ordinary sheet of paper. They can provide 1600 thru 6250 BPI capability and may be used for disk backup as well as data interchange. Complete subsystem prices, including software, start as low as $2,495 for 7" units and at $3,670 for 10½" units.

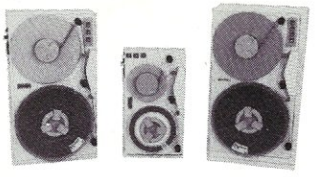
Qualstar has become the market leader in desk-top 9-track systems for a good reason; our tape drives have established an outstanding record for reliability and low cost of ownership.

Discover the many advantages 9-track tape has over other Micro/Mainframe links.

Call us today!

QUALSTAR®

9621 Irondale Avenue,
Chatsworth, CA 91311
Telephone: (818) 882-5822



Macintosh is a trademark of Apple Computer, Inc.

IS NOTHING SACRED?

Now the FULL source code for TURBO Pascal is available for the IBM-PC! WHAT, you are still trying to debug without source code? But why? Source Code Generators (SCG's) provide **completely** commented and labeled ASCII source files which can be edited and assembled and UNDERSTOOD!

SCG's are available for the following products:

__ TURBO Pascal ver 3 (IBM-PC)* . .$67.50
__ TURBO Pascal ver 3 (Z-80)* $45.00
__ CP/M 2.2 . . . . . .$45.00
__ CP/M 3 . . . . . .$75.00
*A fast assembler is included free!

The following are general purpose disassemblers:

__ Masterful Disassembler (Z-80) . . .$45.00
__ Masterful Disassembler (IBM-PC) .$47.50
__ UNREL (relocatable files) (8080) .$45.00

"The darndest thing I ever did see . . ."
Pournelle, BYTE

"I have seen the original source and yours is much better!"
Anonymous, SOG VI

VISA/MC/check Shipping/Handling $1.50
card #_____ Tax $ ____
expires ____/____ Total $ ____

All products are fully guaranteed. Disk format, 8" ☐ 5" ☐ type_____.

C.C. Software, 1907 Alvarado Ave., Walnut Creek, CA 94596, (415) 939-8153

CP/M and TURBO Pascal are trademarks of Digital Research & Borland Int.

PRODUCTS YOU SHOULD KNOW ABOUT!!!

HOT PRINT—*$99.00*
One of the best selling printing utilities for Novell's Netware. Take care of all your spooling needs from within your application. 20k resident. Send messages, display queue, print across servers with auto attach and auto login, un-erase files, display any file, send any printer control codes, download fonts, improve password security, universal print codes, dedicated print server program, plus hundreds of other features and also includes 15 other useful utility programs. Menu or command driven. Create all your own spooling menus or call Hot Print from applications such as spreadsheets, data base systems, menus, batch files, etc.

NEW HOT SERVER PAK—*$99.00*
A new RESIDENT print server. 3k per shared local printer, non-dedicated. Share local printers in the background. Even share multiple local printers. User can still work on workstation. Patch available so no supervisor rights are required! Drives parallel or serial. You specify the speed of the output. You specify the priority of the jobs in the queue. You specify how many local printers you wish to share. Works with Spool/Endspool, no new commands to learn. Also works with Hot Print to offer a complete printing solution on all versions of Netware (2.0x). Also included: HPSERVER—a dedicated print server that supports up to 15 printers on a single workstation and a few other utilities.

NEW HOT LOOK—*$99.00*
A new RESIDENT utility that allows you to view and insert keystrokes into another user's terminal right from your keyboard. Type HL and select the user from a screen full of user names or stations. Pop up their screen and see what's happening. User still maintains control of keyboard, but you can enter keystrokes for them. Start a batch file remotely. Diagnose problems without leaving your desk. Dial up the network and Hot Look over to the work station with the problem and diagnose it from home or the office. Comparable products cost 5 times as much.

Order today. The three pack is just $270.00, or order them individually.
Dealer pricing is available.
Coming soon . . . By the time you read this ad, support for Netware 2.1 as well.

***Call NOW!* Ron Turley Computer Assoc/Hot Ware (602) 581-2447 Telex: 6503471847 MCI UW**

nate any unreasonable delays in the potential printer backlog.

These two decisions left us with the task of creating our own routines to handle printer control through the network. We had obtained the Netware function calls from Novell, along with their routines for using the function calls. When we began examining the material received from Novell, we discovered that the routines were in assembly language and the Pascal portions used external calls to the assembly language routines. What made this disheartening was that there was virtually no documentation on how to use the code provided, and the person who wrote the Pascal routines was no longer available. The last straw was that the assembly language routines were not compiled to object code, and to compile them required an assembler two or three versions beyond the one we had. Thus, we decided to write our own routines so we could have total control of the spooler control routines.

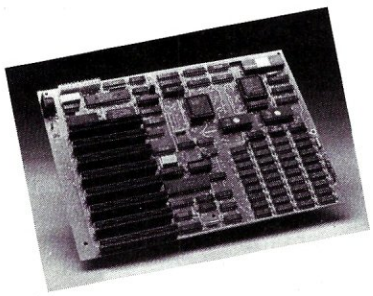
We decided to initially create routines to: set the spooler flags, obtain the printer status, modify the current spooler device, and close a spooler device. With these four routines we could output data to any printer on the network or to local printers at any point in an application. The following sections provide a discussion of each of these routines.

Listing 1. Set_Spool_Flags4

```
procedure Set_Spool_Flags4(PrntNo : byte);
{
This is a netware 4.0 function call and will set the
network printer spooler spool flags.

 Process for outputing:
   start with LST device closed (local)

   Modify_LST_Device
   Set_Spool_Flags(N)
     Perform output
   Close_LST_Device
   repeat for new output
}

type
  {Request Buffer Type}
  ReqBuffType = record
    PacketLength : integer;       {Integer length of the
                                   request buffer}
    SetFlags     : byte;          {$02 Set Spool Flags}
    PrntFlags    : byte;          {Print Control Flags
                                   $08 = Suppress auto
                                         form feed at
                                         end of print job
                                   $20 = Delete spool
                                         file after
                                         printing
                                   $40 = enable tab
                                         expansion
                                   $80 = Print banner
                                         page}
    TabSize      : byte;          {Tab size, 1 - 20}
    TargetPrinter: byte;          {Output printer
                                   number, 1 - n}
    NumCopies    : byte;          {Number of copies,
                                   0 - 255
                                   0 = no printing}
    FormType     : byte;          {Form Type, 0 - 255}
    BannerTxt    : array[1..13] of char; {Banner
                                  Heading, not used here}
    Terminator   : byte;          {Null terminator for
                                   buffer, $00}
  end;

  {8086 register type}
  RegType = record
    case integer of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH            :
byte);
  end;

var
  ReqBuff : ReqBuffType;
  Regs    : RegType;

begin {Set_Spool_Flags4}
  fillchar(ReqBuff,sizeof(ReqBuff),0); {Fill Request
                                        Buffer with nulls}
  with ReqBuff do
  begin
    PacketLength:=$0006;          {work with first 6
                                   bytes of buffer}
    SetFlags:=$02;                {initialize SetFlags}
    PrntFlags:=$28;               {Suppress auto form
                                   feed and delete spool
                                   file after printing}
    TabSize:=1;                   {Tab size = 1
                                   character}
    TargetPrinter:=PrntNo;        {Set network printer
                                   number}
    NumCopies:=1;                 {Number of copies = 1}
  end;

  Regs.AH:=$E0;                   {$E0 - Set spool
                                   flags}

  Regs.DS:=seg(ReqBuff);          {DS:SI address of
                                   request buffer}
  Regs.SI:=ofs(ReqBuff);

  Regs.ES:=Regs.DS;               {ES:DI address of
                                   reply buffer}
  Regs.DI:=Regs.DS;
```


OEMs, System Integrators
Clone Builders . . .

286 BOARDS: $149 or LESS FOR QUANTITES OF 40/MONTH



Provide Your Customers With Over 5 Times The Power of 8 MHz 8088-2 Turbo XTs

The high performance Bullet-286 E motherboard is a proven performance and quality winner world-wide. Now you can use this 80286 compatability and value leader in your own system products at the most favorable prices anywhere. Buy at quantity prices, or enjoy discounts to 79% as a co-op manufacturer under our very attractive inventory, credit, and repurchase terms. [List price with zero RAM $495. One unit for evaluation: $229.]

WAVE MATE, INC.
2341 205th Street, #110 Torrance, CA 90501
Phone: 213-553-8190 Fax: 213-533-5940

```
  intr($21,Regs);                    {perform call}
end; {Set_Spool_Flags4}
```

Listing 2. Set_Spool_Flags2

```
procedure Set_Spool_Flags2(PrntNo : byte);
{
  This procedure is an advanced netware 2.0a call and
will turn off the banner page in spool defaults and set
the printer number.

  Process for outputing:
    with LST device closed (local)

    Set_Spool_Flags(N)
    Modify_LST_Device
      Perform output
    Close_LST_Device
}

type
  RequestBuffer = record
    Status     : byte;
    PrintFlag  : byte;
    TabSize    : byte;
    TargetPrint: byte;
    NumCopies  : byte;
    FormType   : byte;
    BannerText : array[1..13] of char;
    LocalPrint : byte;
    TimeOut    : integer;
    DeviceClose: byte;
  end;
  {8086 register type}
  RegType = record
    case integer of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH            :
byte);
  end;

var
  Reg         : RegType;       {8086 type registers}
  ReqBuff     : RequestBuffer; {Parameters buffer}

begin  {Set_Spool_Flags2}
  {Initialize buffers}
  fillchar(ReqBuff,sizeof(ReqBuff),0);
  with ReqBuff do
  begin
    Status:=0; {enable defaults}
    TabSize:=1;
    TargetPrint:=PrntNo;
    TimeOut:=0;
    PrintFlag:=28; {suppress ff, delete spool file after
                    printing}
  end;

  {Setup the call}
  Reg.Ah:=$B8; {Network call function - Print Function}
  Reg.CX:=$04; {Change only first 4 bytes in request}

  Reg.ES:=seg(ReqBuff); {ES:BX = address of ReqBuff}
  Reg.BX:=ofs(ReqBuff);
  Reg.AL:=$01; {Network subfunction call - Set
                Defaults}

  intr($21,Reg); {perform the call}
end;  {Set_Spool_Flags2}
```

Listing 3. Modify_LST_Device

```
function Modify_LST_Device:byte;
{
  This function will set the Lst device to the network
printer and return the byte $00 on completion.
}

type
  {8086 register type}
  RegType = record
    case integer of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH            :
byte);
  end;
```

... *continued on next page*

Setting the Spooler Control Flags

The spooler control flags are designed to control several spooler options, including banner page printing, deleting the spool file after printing, automatic form feed after printing, and number of copies, among others. There are two routines included to perform these tasks, as shown in Listing 1. (Netware 4.0 Function *$E0*, subsection *$02*, where the *$* indicates hexadecimal values) and Listing 2 (Advanced Netware 2.0 function *$B8*).

The main differences between these two function calls are that the 4.0 call can only set the flags, but it cannot read the current values and cannot set the *TimeOut*, *Local Device*, or *Device Close* values available in Advanced Netware. In addition, there is a difference in the usage of the two routines. With the Netware 4.0 function, you must set the list device, set the spool flags, output the data, and then close the list device. When using the Advanced Netware 2.0 function, however, you must set your spool flags first, then modify your list device, output the data, and close the list device. Modifying the list device before setting the spool flags with the Advanced Netware function will result in unreliable printer settings. This is because the Advanced Netware function requires a call to *Modify LST Device* to actually implement the settings.

The Netware 4.0 Set Spool Flags function *$E0, $02* (Listing 1) provides the ability to set, but not read, several spooler related parameters. These include: suppressing the auto form feed after each print job, deleting the spool file after printing, enabling expansion of tab characters, printing a banner page at the beginning of each job, setting the tab size, selecting a network printer, indicating the number of printed copies, setting the form type, and determining the banner text to be used on the banner pages. These parameters are designated in the variables in the record *ReqBuff*.

The routine provided by Novell does not use the banner text option. If you want a banner page printed at the beginning of your print job you must put the banner text into the variable *BannerTxt*, add the number of text bytes to *PacketLength*, and add *$80* to the *PrntFlags* value to enable banner printing. The text you set in *BannerTxt* must be a null-terminated text string. We also set the variable *PrntFlags* to *$28*, which disables the auto form feed after the print job and enables deletion of the spool file after printing. If you want to use the other *PrntFlags* options you need to add (bit wise OR) the values listed in the record type definition together to obtain the options you want.

The second routine for setting the spool flags uses the Advanced Netware 2.0 function *$B8 Get/Set Default Spool Flags* (Listing 2). We recommend that this newer version call be used, since Novell stated that in the future the older Netware 4.0 calls may not be supported. As mentioned above, this function can set and retrieve the spooler flag values. In addition to retrieving the flags, function *$B8* provides three additional features: indicat-

ing which local port is to be spooled (*LocalPrint*), specifying the time until an automatic *EndSpool* is to be performed (*TimeOut*), and indicating if an automatic *EndSpool* is to be performed (*DeviceClose*). *LocalPrint* can have a value of 0 for LPT1:, 1 for LPT2:, or 2 for LPT3:. All output directed to the specified local port will be spooled to the network printer. *TimeOut* specifies the number of ticks (one tick equals 1/18th second) for the network to wait for the next output before it closes the print job. Lastly, *DeviceClose* should be zero if an automatic *EndSpool* is to be performed when the device is closed (DOS 3.0, 3.1, and 3.2 only). If *DeviceClose* is nonzero, the *EndSpool* will not be automatically performed when the device is closed. You must call this routine prior to calling *Modify_LST_Device.*

Diverting Printer Output

Modify_LST_Device (Listing 3) is used to divert output from the local port to the network printer number specified in the call to *Set_Spool_Flags.* The routine uses Netware function *$DF, $00 Modify_LST_Device* to enable the capture of output information. If the call is successful, *Modify_LST_Device* will return *$00.*

When you have completed a print sequence you will need to cancel the spooling of additional output. The routine *Close_LST_Device* is used to cancel the spooler link established with *Modify_LST_Device.* In *Close_LST_Device,* Netware function *$DF, $01* (Listing 4) is used to perform the closure. If you used the Advanced *Netware Set_Spool_Flags* function and chose to have the device closed after a timeout, you won't need to perform a *Close_LST_Device.* However, if your application is going to call another *Set_Spool_Flags,* you should close the current print job before beginning the new print job. If you are using the Netware 4.0 function you will need to call *Close_LST_Device* before changing devices or finishing your calling routine.

Setting Printer Status

The last routine obtains the status of a network printer. The routine *Get_NPrinter_Status* (Listing 5) provides the ability to determine: if the printer has been halted through the network console; if the printer is off-line; the current form type; and the current printer number. The printer number returned indicates the network printer number that will ultimately receive the output. This may be a different printer than the requested printer number, if the requested printer has been rerouted to a different printer on the network by the network console.

Prior to Standard Netware and Advanced Netware, Version 2.1, have a problem in obtaining these status values. With these older versions you can only obtain the status values if a print job is waiting to be printed or is in the process of being printed. Novell sources say this problem has been corrected in Advanced Netware Version 2.1. However, Netware 2.1 is currently available only for Netware 286 systems. If you are using Netware 86, you will need to wait for Version 2.1.

... continued

```
var
  Reg : RegType; {8086 type registers}

begin  {Modify_LST_Device}

  {Setup the call}
  Reg.AH:=$DF; {Network call function - Print Function}
  Reg.DL:=$00; {Network subfunction call - Start Lst
                catch}

  intr($21,Reg); {Perform the interrupt}

  Modify_LST_Device:=Reg.Al; {return result}
end;  {Modify_LST_Device}
```

Listing 4. Close_LST_Device

```
function Close_LST_Device:byte;
{
  This function will close the catch and queue for
printing and return the byte $00 on completion.
}

type
  {8086 register type}
  RegType = record
    case integer of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH            :
byte);
  end;

var
  Reg : RegType; {8086 type registers}

begin  {Close_LST_Device}
  {Setup the call}
  Reg.Ah:=$DF; {Network call function - Print Function}
  Reg.Dl:=$01; {Network subfunction call - Start Lst
                catch}

  intr($21,Reg); {Perform the interrupt}

  Close_LST_Device:=Reg.Al; {return result}
end;  {Close_LST_Device}
```

Listing 5. Get_NPrinter_Status

```
function Get_NPrinter_Status(PrinterNo : byte) :
boolean;
{
  This function will get the status of the server
printer specified by PrinterNo and will return true if
the printer is On-line and not halted, or false if it
is Off-line or halted.  PrinterHalted will be $FF if
the network console halted the server printer.
PrintFormType - will contain the forms type number
currently mounted in the indicated printer.
TargetPrinter - will indicate the routed printer, this
number will be the same as PrinterNo unless the network
console has rerouted the printer.
  Note: This function will only return false in
advanced netware 2.0a and earlier versions when the
printer queue has contents and the printer is off line.
If there are no jobs pending the function will return
an On-line status.
}

type
  {Request buffer format}
  ReqBuffType = record
    PacketLength : integer;
    Command      : byte;
    PrntNo       : byte;
  end;

  {Reply buffer format}
  RepBuffType = record
    PacketLength   : integer;
    PrinterHalted  : byte;
    PrinterOffLine : byte;
    PrinterFormType: byte;
    TargetPrinter  : byte;
  end;
```

Summing Up

These routines were developed in Turbo Pascal and designed to accommodate our applications. There are several options and variations that can be accomplished with these routines to meet your specific needs. Listing 6 is simple example of how to use the new and old *Set_Spool_Flags, Modify_LST_Device,* and *Close_LST_Device* to enable/disable printers and switch between printers.

These routines are only a few of the many functions available to systems integrators working on Novell Networks. You can obtain a complete list of the function calls from Novell by becoming a Netware Affiliate. If you would like information about becoming an affiliate, you should contact Kim Perry or David Hills at Novell, Inc., Mail Stop #BA2MS11, 122 East 1700 South, Provo, Utah 84601; (801) 379-7508. The Affiliate Program is open to most dealers and software developers who want to develop or support software on Novell Networks. The affiliation is free but there is an agreement that must be completed prior to receiving information. □

Ed Rought and Tom Hoops are co-owners of Perpetual Data Systems, a software development and marketing firm located at 63 Keystone Ave., Ste. 206, Reno, NV 89503. Along with their commercial installations, they are the co-authors of a series of books on Pascal and C routines for software application developers to be released shortly by Howard W. Sams & Co., Inc.

```
  {8086 register type}
  RegType = record
    case integer of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags :
integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH            :
byte);
  end;

var
  ReqBuff : ReqBuffType;
  RepBuff : RepBuffType;
  Regs    : RegType;

begin {Get_NPrinter_Status}
  fillchar(ReqBuff,sizeof(ReqBuff),0); {null out
                                        Request buff}
  fillchar(RepBuff,sizeof(RepBuff),0); {null out Reply
                                        buffer}

  with ReqBuff do
  begin
    PacketLength:=sizeof(ReqBuff)-2; {Length of buffer
                                      info}
    Command:=$06; {Get printer Status}
    PrntNo:=PrinterNo; {get status for network printer
                        # PrntNo}
  end;

  RepBuff.PacketLength:=sizeof(RepBuff)-2; {Length of
                                     Reply buffer info}

  Regs.AH:=$E0; {Netware spooling functions}

  Regs.DS:=seg(ReqBuff); {DS:SI - address of request
                          buffer}
  Regs.SI:=ofs(ReqBuff);
  Regs.ES:=seg(RepBuff); {ES:DI - address of reply
                          buffer}
  Regs.DI:=ofs(RepBuff);

  intr($21,Regs); {do the call}

  {analyze results}
  with RepBuff do
  begin
    if (RepBuff.PrinterHalted<>$00) or
       (RepBuff.PrinterOffLine<>$00) or
       (Regs.AL<>$00) then
      Get_NPrinter_Status:=false
    else
      Get_NPrinter_Status:=true;
  end;
end; {Get_NPrinter_Status}
```

Listing 6. Sample Usage Program

```
program Spooling_Demo;

{Include the spooler functions at this point}

begin
  {Set to local mode to start with}
  if Close_LST_Device=0 then;  {Do it blind}

  writeln(lst,'Here I am on the local printer');

  {Change to network printer}
  if Modify_LST_Device<>0 then {Note: could do these
                                blind too.}
  begin
    writeln('Could''nt Modify List device.');
    halt;
  end;

  {Set up network printer & select printer 0}
  Set_Spool_Flags4(0);

  writeln(lst,'Here I am on the network printer 0.');

  if Close_LST_Device=0 then;  {go back to local
                                printer & spool (blind)}

  writeln(lst,'Here I am on the local printer again');

  {Change to network printer}
  if Modify_LST_Device<>0 then
  begin
    writeln('Could''nt Modify List device.');
    halt;
  end;

  {Set up network printer & select printer 1}
  Set_Spool_Flags4(1);

  writeln(lst,'Here I am on the network printer 1.');

  if Close_LST_Device=0 then;  {go back to local
                                printer & spool (blind)}

  {Now using the advanced netware 2.0a call}

  writeln(lst,'Here I am on your local printer for the
last time.');

  {Set up network printer & select printer 0}
  Set_Spool_Flags2(0);

  {Go from local to spool}
  if Modify_LST_Device<>0 then
  begin
    writeln('Could''nt Modify List device.');
    halt;
  end;

    writeln(lst,'Here I am on Network printer 0
      thanks to 2.0a');

  if Close_LST_Device=0 then;  {go back to local printer
                                & spool (blind)}
end.
```

MicroWay's RTOS

A Port of Intel's Real-Time Operating System to the PC

by Stuart Jones, M.D.

Intel is best known as the developer and manufacturer of the 8088, 80286, and 80386 chips—the chips that power IBM and IBM-compatible PCs. The company's reputation for publishing 8- and 16-bit operating systems and language compilers is known mainly to those in the industrial microcomputer arena, and is unknown to most PC owners.

Intel's potent real-time multitasking operating system for 808X-series CPUs, RMX-86, was released in 1980. RMX has been most popular for industrial grade microsystems. These Multibus I- and II-based machines are commonly used for robotics, process control, and laboratory data collection—applications requiring systems software that permits the machine to respond quickly to and synchronize with external events. Two such computers known to this author are used to collect Nuclear Medicine image data from radiation detectors known as gamma cameras. Each gamma ray detected must be processed individually, at very high count rates. Both of these 8-MHz 8086 systems (which can handle up to 80,000 counts per second) use RMX-86. In one of them, currently used in the author's laboratory, RMX manages the operations of three 8086 processors sharing a common Multibus.

RMX-86 is a mature product (the current release is Version 6). Recent versions will run on 8086, 8088, and 80186 processors, and also on the 80286 in real mode; and Intel has just announced RMX-286, which operates in protected mode on the 80286. The company supplies these operating systems both preconfigured (for Intel's own development computer systems) and in user-configurable form, complete with source code. Given the popularity and low cost (relative to industrial systems) of PCs, it was inevitable that RMX would be ported to IBM-compatible equipment.

MicroWay, a company specializing in high-performance hardware and software products, has offered a PC- and PC/XT-compatible version of RMX-86 for several years and has just released an RTOS version configured for the 6-MHz PC/AT. Given the current interest in concurrency for PCs, a close look at a product that legitimately claims to be the first real-time multitasking operating system for IBM-standard micros is both timely and appropriate.

Reviewing an operating system, however, is a job almost ridiculous in its enormity. I admit to intense feelings of inadequacy in confronting the dauntingly large pile of documentation currently occupying a prominent position in my computer workspace. Although I use an RMX-based computer system for Nuclear Medicine every day, RTOS is the first programmable RMX version I have used. In the time allotted, it was impossible to explore the nooks and crannies of the RMX system structure, so rather than an in-depth review, I will try to describe the most important features of RMX, and to distill the flavor of the RTOS version for those interested in exploring further. Specifically, I will try to address:

1. The capabilities of RMX that make it interesting to the prospective application developer;
2. The difficulty of learning RMX (that is, its documentation quality) and getting the RTOS version running on a PC or AT;
3. The ease (or lack thereof) of developing software under RMX, given the tools distributed with RTOS; and
4. The usability and performance of the multitasking features of RTOS.

Because of the size and complexity of RMX, this review will be published in two parts. This first part will explore the generic features and facilities of RMX—those found in any preconfigured version. The second part to be published later this year will examine the peculiarities of the MicroWay implementation, as well as offering some hands-on experience and benchmarks.

The Structure of RMX-86

As operating systems (OS) go, RMX is relatively complicated—it is more complicated than DOS, but not nearly as complicated as many minicomputer or mainframe systems.

Like DOS, RMX is put together as a series of layers (Figure 1). Each layer may call upon services provided by those below it.

The bottom layer, the Nucleus, manages task prioritizing, scheduling, and synchronization, as well as intertask communication. It also manages all hardware and software interrupts, including those from the system's clock-timer.

The next two layers up are the Basic I/O System (BIOS) and the Extended I/O System (EIOS). These manage mass-storage data files in a device-independent manner. Specific hardware is supported via drivers (as in other OSs), though only two of these—for the system keyboard and display—must be preconfigured into the BIOS. The Application Loader, which manages transfer of programs and program overlays between mass storage and main memory, is considered to be part of the EIOS.

Above the EIOS sits the Human Interface, which handles interactions between the lower levels of RMX and terminal devices. This layer implements the RMX file and directory structure on mass storage. The Human Interface provides facilities for multiuser operation of RMX via separate terminals, complete with user numbers, separate user main memory workspaces, and file directories, file access security, and so on.

The next-to-last layer up is the Universal Development Interface (UDI). This provides system calls for file access, system date and time, and memory allocation—services also provided by lower levels. The function of the UDI is to hide the details involved with the lower-level system calls, and thus facilitate application development. It is possible to develop a UDI that runs under a non-RMX, 808X operating system. This permits a non-real-time/multitasking RMX application (e.g., an editor, compiler, or link-editor) to run under the foreign OS. In fact, such a UDI for DOS is supplied by MicroWay as an adjunct to RTOS. (I suspect that facilitating cross development on single-user systems is a major reason for the UDI.)

The topmost layer consists of all user-written applications, as well as the utility programs provided by Intel for RMX, and RMX's program development facilities.

Resource Management and Jobs

RMX manages system resources, including tasks, as objects. Each is assigned an identifying, unsigned, 16-bit token or identifier at the time of creation or definition. Tokens may be passed freely among tasks, giving the software designer considerable latitude.

In addition to tasks, other objects recognized by RMX include memory segments; semaphores, mailboxes, and regions (used for intertask communication and synchronization); files on mass storage; and jobs. Composite objects can be created from these categories. It is also possible (although very ambitious) to create both extension objects and the system services necessary to manipulate them.

A job consists of one or more tasks, with associated other objects, a memory pool (assigned when the job is created), and an object directory. When the Human Interface initializes during the RMX bootstrap process, it creates an interactive job for each terminal. This becomes the root job for that console; all other jobs (e.g., applications programs) run thereafter are descendants of the root job, and therefore draw memory from its memory pool. Since current RMX versions do not offer virtual memory support, this limits the practical number of simultaneous users on an 808X computer to three or four.

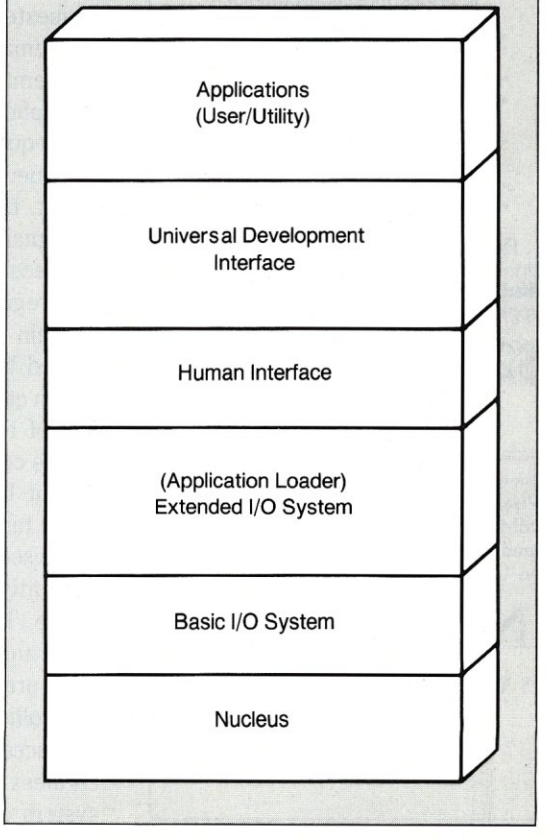
Each job has a root task that creates any other tasks (and jobs) related to the job. Such descendant tasks can be passed other resources at the time they are created, either as messages through mailboxes or by looking up their own tokens in the object directory. Each job is assigned a maximum priority level for any of its tasks at the time the job is created.

Task Switching and Synchronization

Each task is given a priority level ranging from 0 (highest) to 255 (lowest). Levels 0–127 are reserved for interrupt handlers and interrupt tasks. Task priority is assigned at the time of creation and may be subsequently altered by the task itself or by another task possessing its token. The maximum priority that can be assigned to a task is limited by that of its parent job. This limit does not apply to interrupt-related tasks.

A given task occupies one of five possible states: (1) running, (2) ready, (3) suspended, (4) asleep, or (5) asleep-suspended. A suspended task is memory resident, but it can-

Figure 1. RMX System Diagram



Introducing

NANODISK

"Disk Cache for the IBM PC"

Make your floppy drive and hard disk run close to RAM disk speeds. Dramatic speed improvement for most programs. Supports cache of any size in main or expanded memory.

Requires IBM PC/XT/AT or true clone.

only $29.95

MultiDos Plus

"multitasking for the IBM-PC."

Ideal for developing applications in process control, data acquisition, communications, and other areas. Check these features which make **MultiDos Plus** an unbeatable value.

- Run up to 32 DOS programs concurrently.
- Operator commands to load/run programs, change priority, check program status, abort/suspend/resume programs.
- Programmatic interface via INT 15H for the following.
  - * Intertask message communication.
  - * Task control by means of semaphores.
  - * 256 priority levels.
  - * Suspend task for specified interval.
  - * Spawn and terminate external and internal tasks.
  - * Disable/enable multitasking.
  - * and more!

Requires IBM PC/XT/AT or true clone, and enough memory to hold **MultiDos Plus** (48 KB) and all your application programs.

$24.95 *or* $99.95

With source code (Written in Lattice C and Microsoft Assembler.)

Outside USA add $5.00 shipping and handling.
Visa and Mastercard orders only call toll-free: 1-800-872-4566, ext. 350., or send check or money order (Drawn on U.S. Bank Only) to:

NANOSOFT

13 Westfield Rd, Natick, MA 01760
MA orders add 5% sales tax.

not run until it is resumed by another task or an associated interrupt. Task suspensions can be nested; each nested suspension must be counteracted by a resumption before the task can again become ready. An asleep task is either waiting for a request to be granted, or has put itself to sleep for a specified amount of time via a system call. An asleep task can also be suspended (thus entering the asleep-suspended state); if its request is granted or its sleeping time expires before it is resumed, the task then enters the suspended state.

As RMX is distributed by Intel, task switching is entirely event-driven; that is, there is no inherent time-slicing mechanism to apportion CPU time between tasks. Instead, task switching will occur only if a higher-priority task becomes ready, or if the current task goes to sleep, suspends itself, or deletes itself. However, specific RMX configurations typically do implement time slicing via the system clock-timer interrupt service routine.

Task synchronization is handled using the semaphore and region objects. The semaphore has an associated queue of tasks. Each task has requested one or more "units" from the semaphore. The units are sent to the semaphore by other tasks. If the semaphore has enough units to fulfill the request of the task at the head of the queue, that task becomes ready. If not, that task (and others behind it) remains asleep until enough units are received.

A region guards access to the area of main memory used to store data shared between tasks. It also manages a queue of tasks waiting for control of the region. Once a task receives control, it becomes ready and cannot be suspended or deleted by other tasks; its priority is temporarily raised to equal that of the task currently at the head of the "ready" queue. This ensures that the shared data cannot be corrupted by inadvertent pre-emption of the currently controlling task by another task that has access to the data. However, careless use of regions can lead to system deadlock. It also can represent a security risk on multiuser RMX installations if it creates a task that cannot be deleted, even when a user logs off. Therefore, Intel recommends that regions be used only as part of extension objects (wherein such use can be controlled by system programmers), or by tasks directly involved in RMX operation (i.e., not involving the Human Interface).

Intertask communication is managed using the mailbox object. A mailbox manages two queues: one consisting of tasks waiting for messages and the other of messages waiting for tasks. Each message is a token and messages in the queue are passed to tasks on a FIFO basis.

For semaphores, regions, and mailboxes, the queue of waiting tasks may be arranged either in order of task priority or in order of arrival (i.e., FIFO). The ordering mechanism used is selected at the time the semaphore, mailbox, or region is created.

Tasks requesting access to a region, a message from a mailbox, or units from a semaphore may either wait for the request to be filled (thus entering the asleep state) or continue if the request cannot be filled, receiving an exception code if the request was not honored.

Interrupt Handlers and Tasks

The RMX-86 Nucleus provides extensive services for managing hardware and software interrupts. A user-designed interrupt handler may be bound to any of the 808X/80X86 interrupt levels, although certain levels (particularly that assigned to the system clock) are reserved by RMX. When the handler is assigned, an interrupt task that finishes the interrupt processing begun by the handler may also be assigned to the same or a different level—the handler passes control to the interrupt task by asserting a software interrupt. RMX assigns a default priority, equal to double the level number, to an interrupt handler, or task according to the interrupt level to which it is assigned for interrupt levels

0–3FH. Thus, every even priority level from 2 to 128 is preassigned.

Interrupt handlers are limited in terms of the system calls they may invoke, and also (practically speaking) in the amount of processing time they can tie up, since other lower-priority interrupts are disabled while the handler is running. An interrupt task, however, can continue processing while other interrupts are serviced, and can invoke services (e.g., I/O calls) forbidden to handlers. Invocations of interrupt tasks can be nested up to a maximum count, which is defined when the interrupt handler is bound.

Exceptions

As befits an OS of such complexity, RMX provides facilities for managing situations where something has gone wrong, either due to programmer error or to an environmental problem that could not reasonably have been anticipated. As each task or job is created, it is assigned (by default) or assigns itself a procedure for managing exceptions. This not only provides for robust and context-sensitive error handling, but also represents a mechanism for circumventing Intel- or RMX-configurator-imposed limitations (since an exception handler can choose to ignore certain exception conditions). A job that does not assign its own exception handler inherits that of its parent job (e.g., the system exception handler, where no other job has specified otherwise). If a job is assigned a different exception handler, this becomes the default for all descendant tasks or jobs.

In addition to its exception handler, a task or job is assigned an exception mode. This defines when control will be passed to the exception handler—for every exception, for no exception (effectively disabling the handler), for programmer errors only, or for environmental conditions only. The choice of exception handler or mode can be reassigned via system calls.

In addition, most RMX system calls can return an exception code, which is returned to the calling task (and which may be passed to the exception handler, if it is enabled to manage the condition being signaled). Certain of these codes are informational only, and will not invoke a handler.

The I/O System

I/O services are provided via system calls to the BIOS and the EIOS (or to the UDI, if desired). BIOS system calls require specification of a number of details (e.g., buffer addresses and sizes) that are routinely provided for by EIOS calls. However, BIOS calls typically involve less overhead than EIOS calls, and may allow processing to proceed concurrently with I/O. EIOS calls put the calling task to sleep until I/O is complete. However, the EIOS does permit assignment of logical names to physical files and devices, easing the programmer's burden. As mentioned above, UDI calls can be used to provide limited, stereotyped I/O services for user applications that may need to run in an 808X cross-development environment (precluding, as well, the use of RMX real-time and multitasking capabilities by such applications).

The BIOS incorporates mechanisms for I/O with named files on mass storage, physical files (e.g., line printers), and stream files (temporary mass storage files used for passing data from the output of one task to the input of another, similar to pipes in UNIX). The BIOS implements communication with these files in a device-independent manner. As for most other OSs, the grubby details of getting the data to and from physical media are managed by device drivers.

An unusual feature of RMX is its dynamic device configuration. Using utility programs, different device drivers may be attached to or detached from the BIOS while RMX is running. Indeed, the system device may be dynamically respecified (although this does involve rebooting) if the supplied bootstrap mechanism or a supplied utility program provides for this. As mentioned above, the only devices that need to be

COMBINE THE
RAW POWER OF FORTH
WITH THE CONVENIENCE
OF CONVENTIONAL LANGUAGES

HS/FORTH

Yes, Forth gives you total control of your computer, but only HS/FORTH gives you implemented functionality so you aren't left hanging with "great possibilities" (and lots of work!) With over 1500 functions you are almost done before you start!

WELCOME TO HS/FORTH, where megabyte programs compile at 10,000 lines per minute, and execute faster than ones built in 64k limited systems. Then use AUTOOPT to reach within a few percent of full assembler performance — not a native code compiler linking only simple code primitives, but a full recursive descent optimizer with almost all of HS/FORTH as a useable resource. Both optimizer and assembler can create independent programs as well as code primitives. The metacompiler creates threaded systems from a few hundred bytes to as large as required, and can produce ANY threading scheme. And the entire system can be saved, sealed, or turnkeyed for distribution either on disk or in ROM (with or without BIOS).

HS/FORTH is a first class application development and implementation system. You can exploit all of DOS (commands, functions, even shelled programs) and link to .OBJ and .LIB files meant for other languages *interactively!*

I/O is easier than in Pascal or Basic, but much more powerful — whether you need parsing, formatting, or random access. Send display output through DOS, BIOS, or direct to video memory. Windows organize both text and graphics display, and greatly enhance both time slice and round robin multitasking utilities. Math facilities include both software and hardware floating point plus an 18 digit integer (finance) extension and fast arrays for all data types. Hardware floating point covers the full range of trig, hyper and transcendental math including complex.

Undeniably the most flexible & complete Forth system available, at any price, with no expensive extras to buy later. Compiles 79 & 83 standard programs. Distribute metacompiled tools, or turnkeyed applications royalty free.

HS/FORTH (complete system):	$395.
HS/FORTH: Tutorial & Ref (500 pg)	$ 59.
Forth: Text & Reference (500 pg)	$ 22.
HS/FORTH Glossary	$ 10.
GIGAFORTH Option (Beta release)	$245.

(Native Mode from SOFTMILLS, INC.)

VISA Visa Mastercard MasterCard

HARVARD
SOFTWORKS
PO BOX 69
SPRINGBORO, OH 45066
(513) 748-0390

preconfigured into RMX are those for the default system console.

Terminal I/O and File Management

Console I/O is managed by the Human Interface, which also provides for file access control and directory management, and for running programs via a command line interface (CLI). Individual users are assigned user ID numbers that identify their owned files and directories—a scheme used by most multiuser OSs.

The file structure managed by the Human Interface is hierarchical and quite similar to that provided by UNIX. Every mass storage device is given a logical name when attached (a string expression bracketed by colons—e.g., *:w0:* for the first fixed disk drive, *:sd:* for the default system mass storage device). Each mass storage volume has a root directory with subdirectories, the tree directory structure used by so many operating systems. As for UNIX, user directories are listed in the */USER* directory, with a different subdirectory for each user number. The system manager directory and privileges are password-protected, although the password is initially set to a null string. On a new installation, the system manager must define "terminal definition files" for any additional terminals to be used (in addition to the PC keyboard and display in the case of RTOS), and set up user ID numbers and directories for other users of the terminals. This is unnecessary for RTOS if only the PC keyboard and display will be used for interaction with the system.

As is clearly evident from the system documentation, RMX is designed to run on hardware incorporating at least one fixed disk drive that can serve as the system device (*:sd:*). Devices with removable media (e.g., floppy or cartridge hard drives) must be attached before a volume may be accessed, and detached before that volume is removed. Otherwise, the file structure on that volume and the next one inserted may be corrupted! A volume cannot be removed from the system device without rebooting. (Shades of CP/M-80 V1.4!)

As noted above, on start-up the Human Interface creates an interactive job for each terminal. This job is assigned a specific memory pool (about 225K net under RTOS). Human Interface commands or user programs are limited to the initial memory partition. This can be altered by the system manager, however, if applications require more memory. RTOS applications (including language processors) typically make heavy use of overlays to permit operation in limited memory space.

Intel supplies a comprehensive set of utility programs for managing files, formatting and verifying volumes, setting system date and time, running batch files, and so on. MicroWay supplements these with additional utilities.

The MicroWay Product

The preceding sections are an all-too-brief review of the features and structure of an extremely complicated operating system. In this discussion, I have alluded to some specific RTOS features, more of which will be described in the second part of this review.

RTOS is directly distributed only by MicroWay. Prospective users are warned about possible hardware incompatibilities. At present, RTOS is certified to run only on the 256K-motherboard PC, all 8088-based XTs, and the 6-MHz AT by IBM equipped with the standard monochrome/printer or color graphics display adapter and the IBM fixed-disk controller. The Maynard Inc. add-on hard disk and clock controllers are supported for the PC. With the sole exception of the Compaq 8088-based portable, no other hardware, and no PC/XT/AT clone, is likely to run either RTOS or its companion utility programs. The target PC, XT or AT must have a math coprocessor (8087 or 80287) installed. An RTOS

NEW!

Documentation is a PAIN!

. . . without DocuMotion™

We've found that well indexed and easily accessed documentation is a powerful tool and asset. Now you can simply pop up **DocuMotion** to access, display and print **your** documentation. **DocuMotion** builds indexed document libraries from documentation contained in your source code or any text file.

DocuMotion for programmers:

- Your documentation is available ANYWHERE, ANY TIME.
- Access, display and print your documentation by name or by user-defined categorization trees.
- 19 function Microsoft Windows-style menu bar.
- Powerful print & copy functions.

DocuMotion for project mgrs:

- Programmers produce more and better documentation.
- Reduced function redundancy.
- Greater programmer productivity.

DocuMotion for the PC:

- Runs memory resident or non-resident on any IBM PC/XT/AT or compatible.
- Compatible with all popular memory resident programs.
- Requires only 93K RAM.
- LAN compatible.

DocuMotion is for YOU:

Call NOW 1-612-884-5860

At a special introductory price of ONLY **$159.95** with ANSI 'C' document library.

Demo disk for $10.00 that puts the ANSI 'C' library functions on-line.

NWP - Intelligent Solutions, Inc. P.O. Box 20478 Bloomington, MN. 55420-0478

UNLEASH YOUR 80386!

Your 80386-based PC should run two to three times as fast as your old AT. This speed-up is primarily due to the doubling of the clock speed from 8 to 16 MHz. The new MicroWay products discussed below take advantage of the real power of your 80386, which is actually 4 to 16 times that of the old AT! These new products take advantage of the 32 bit registers and data bus of the 80386 and the Weitek 1167 numeric coprocessor chip set. They include a family of MicroWay 80386 compilers that run in protected mode and numeric coprocessor cards that utilize the Weitek technology.

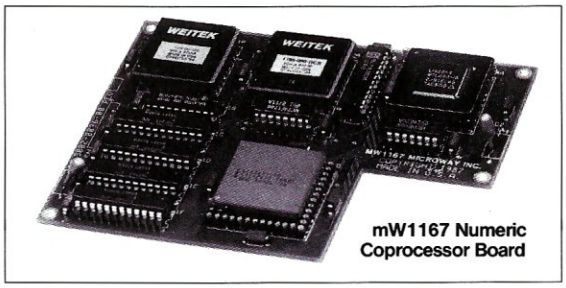
The benefits of our new technologies include:

- An increase in addressable memory from 640K to 4 gigabytes using MS-DOS or Unix.
- A 12 fold increase in the speed of 32 bit integer arithmetic.
- A 4 to 16 fold increase in floating point speed over the 80387/80287 numeric coprocessors.

Equally important, whichever MicroWay product you choose, you can be assured of the same excellent pre- and post-sales support that has made MicroWay the world leader in PC numerics and high performance PC upgrades. For more information, please call the Technical Support Department at

617-746-7341

After July 1988 call 508-746-7341



mW1167 Numeric Coprocessor Board

MicroWay® 80386 Support

MicroWay 80386 Compilers

NDP Fortran-386 and **NDP C-386** are globally optimizing 80386 native code compilers that support a number of Numeric Data Processors, including the 80287, 80387 and mW1167. They generate mainframe quality optimized code and are syntactically and operationally compatible to the Berkeley 4.2 Unix f77 and PCC compilers. MS-DOS specific extensions have been added where necessary to make it easy to port programs written with Microsoft C or Fortran and R/M Fortran.

The compilers are presently available in two formats: Microport Unix 5.3 or MS-DOS as extended by the Phar Lap Tools. MicroWay will port them to other 80386 operating systems such as OS/2 as the need arises and as 80386 versions become available.

The key to addressing more than 640 kbytes is the use of 32-bit integers to address arrays. NDP Fortran-386 generates 32-bit code which executes 3 to 8 times faster than the current generation of 16-bit compilers. There are three elements each of which contributes a factor of 2 to this speed increase: very efficient use of 80386 registers to store 32-bit entities, the use of inline 32-bit arithmetic instead of library calls, and a doubling in the effective utilization of the system data bus.

An example of the benefit of excellent code is a 32-bit matrix multiply. In this benchmark an NDP Fortran-386 program is run against the same program compiled with a 16-bit Fortran. Both programs were run on the same 80386 system. However, the 32-bit code ran 7.5 times faster than the 16-bit code, and 58.5 times faster than the 16-bit code executing on an IBM PC.

NDP FORTRAN-386™ $595
NDP C-386™ $595

MicroWay Numerics

The **mW1167™** is a MicroWay designed high speed numeric coprocessor that works with the 80386. It plugs into a 121 pin "Weitek" socket that is actually a super set of the 80387. This socket is available on a number of motherboards and accelerators including the AT&T 6386, Tandy 4000, Compaq 386/20, Hewlett Packard RS/20 and MicroWay Number Smasher 386. It combines the 64-bit Weitek 1163/64 floating point multiplier/adder with a Weitek/Intel designed "glue chip". The mW1167™ runs at 3.6 MegaWhetstones (compiled with NDP Fortran-386) which is a factor of 16 faster than an AT and 2 to 4 times faster than an 80387.

mW1167 16 MHz $1495
mW1167 20 MHz $1995

Monoputer™ – The INMOS T800-20 Transputer is a 32-bit computer on a chip that features a built-in floating point coprocessor. The T800 can be used to build arbitrarily large parallel processing machines. The Monoputer comes with either the 20 MHz T800 or the T414 (a T800 without the NDP) and includes 2 megabytes of processor memory. Transputer language support from MicroWay includes Occam, C, Fortran, Pascal and Prolog.

Monoputer T414-20 with 2 meg[1] ...$1495
Monoputer T800-20 with 2 meg[1] ...$1995

Quadputer™ can be purchased with 2, 3 or 4 transputers each of which has 1 or 4 megabytes of memory. Quadputers can be cabled together to build arbitrarily fast parallel processing systems that are as fast or faster than today's mainframes. A single T800 is as fast as an 80386/mW1167 combination!

Biputer™ T800/T414 with 2 meg[1] $3495
Quadputer 4 T414-20 with 4 meg[1] ...$6000

[1] Includes Occam

80386 Multi-User Solutions

AT8™ – This intelligent serial controller series is designed to handle 4 to 16 users in a Xenix or Unix environment with as little as 3% degradation in speed. It has been tested and approved by Compaq, Intel, NCR, Zenith, and the Department of Defense for use in high performance 80286 and 80386 Xenix or Unix based multi-user systems.

AT4 – 4 users $795
AT8 – 8 users $995
AT16 – 16 users $1295

Phar Lap™ created the first tools that make it possible to develop 80386 applications which run under MS-DOS yet take advantage of the full power of the 80386. These include an 80386 monitor/loader that runs the 80386 in protected linear address mode, an assembler, linker and debugger. These tools are required for the MS-DOS version of the MicroWay NDP Compilers.

Phar Lap Tools $495

PC/AT ACCELERATORS

287Turbo-10 10 MHz $450
287Turbo-12 12 MHz $550
287TurboPlus-12 12 MHz $629
FASTCACHE-286 9 MHz $299
FASTCACHE-286 12 MHz $399
SUPERCACHE-286 .....$499

MATH COPROCESSORS

80387-20 20 MHz $895
80387-16 16 MHz $495
80287-10 10 MHz $349
80287-8 8 MHz $259
80287-6 6 MHz $179
8087-2 8 MHz $154
8087 5 MHz $99

MicroWay

The World Leader in PC Numerics

P.O. Box 79, Kingston, Mass. 02364 USA (617) 746-7341
32 High St., Kingston-Upon-Thames, U.K., 01-541-5466
St. Leonards, NSW, Australia 02-439-8400

version for the 8-MHz PC/AT and clones is not expected to be ready for 4–6 months. Microway has a current user base of more than 500 installations (nearly all on XTs, but with a growing number of ATs) and appears committed to supporting and developing this product.

The base RTOS package includes the configured RMX with all standard utility programs and object code libraries, as well as the ASM-86 assembler, and MicroWay's own 87-DEBUG. Language translators for Pascal (a pretty much ISO standard implementation, according to the documentation), FORTRAN (1977 standard, also an apparently complete implementation) and PL/M-86, as well as MicroWay's own ETX programming editor and the aforementioned MS-DOS-specific UDI, are available as options.

Documentation is supplied in 8½-by-11 inch, three-hole punched form. In addition to their own manual (1½ inches thick, and clearly written, with some specific examples of terminal dialogue and programming of multitasking applications), MicroWay distributes the Intel ASM-86 and Getting Started with RMX-86 manuals. Additional Intel manuals for the Nucleus, UDI, BIOS, and others, are readily available from MicroWay or Intel. The Intel Manuals appear well-organized and clearly written, but are definitely aimed at the technically knowledgeable user.

RTOS is designed to coexist on a fixed disk with DOS. Indeed, the RMX system loader runs only from DOS. This is a minor nuisance if RMX is being used as the principal operating system for a special project, but it is a major advantage if the target machine is to be used for DOS applications as well. Utility programs are provided to move data files between RMX and DOS, as well as for backup of the original RMX-format distribution diskettes under DOS. The system files are compact enough that RTOS can be run using two 320K floppy disk drives. This is neither desirable nor recommended, however, though the MicroWay manual briefly describes how to navigate with this configuration.

Current pricing of this product is very reasonable given the complexity and support requirements of RMX. A complete configuration, with all available languages, the ETX editor, and the DOS UDI, retails for $2,500. □

Dr. Stuart Jones is a Nuclear Physician with 22 years experience working with computers of all types. He is Chief of Nuclear Medicine at Lehigh Valley Hospital Center in Allentown, Pennsylvania.

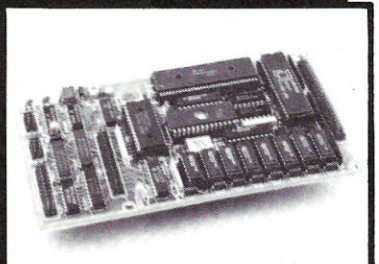
Product Information

Intel
5200 N.E. Elam Young Pkwy.
Hillsboro, OR 97124-0497
(800) 538-3373

MicroWay
P.O. Box 79
Kingston, MA 02364
(617) 746-7341

THE ULTIMATE IN COMPACT CP/M COMPATIBLE COMPUTERS

DSB-8100



Features:
- *Hitachi 64180 CPU running at 6MHz (executes a superset of Z80 instruction set)*
- *256K dynamic RAM*
- *8K EPROM with boot / monitor program standard, up to 32K EPROM optional*
- *1773 Floppy controller supports 40 and 80 track 5¼" and 3½" drives*
- *Host / target SCSI port can use DMA for all transfers*
- *Two RS-232 serial ports support asynchronous communications up to 38,400 baud*
- *Centronics type parallel printer port*
- *CP/M 2.2 optional*
- *Power requirements: +5V at 1.0A, +12V at .05A*
- *Size: 6-3/4" x 3-7/8"*

$365.00 Quantity discounts

Compatible board with 512K RAM and 6 serial ports also available.

Davidge Corporation
P.O. Box 1896
94 Commerce Drive
Buellton, CA 93427
(805) 688-9598

Davidge

• 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 •

NOW!

C / Pascal → 386

Paradox 386
Foxbase+ 386
386-MATLAB,/Weitek
... and others ...

These and other protected-mode 32-bit 80386 programs are among the first to take advantage of the full power of the 386. They and practically every **386 protected-mode** MS-DOS program that's shipping were done with MetaWare's compilers.

It's no surprise. The recognized leader, MetaWare introduced the *first* **C** and **Pascal** compilers that generate protected-mode 386 code for running on any 386 MS-DOS machine (e.g., the Compaq 386 or the IBM PS/2-80) over a year ago. **High C** ™ and **Professional Pascal** ™ are well-established and proven.

Smart software developers aren't waiting! Industry leaders such as Borland (ANSA) and Fox use MetaWare's compilers to get dramatic increases in speed and functionality. Don't wait years for Microsoft's 386DOS—your competition will have a big jump on you!

Expand your application to the large 32-bit address space and the full 32-bit registers of the 80386. Go with the long-standing leader. Contact MetaWare for your 80386 software solution today!

(408)429-6382 telex 493-0879

MetaWare™ INCORPORATED

903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060

The Clear Choice for Large Programming Projects —PC Tech Jrnl

© 1987 MetaWare Trademarks: Paradox 386 Ansa Foxbase+ 386 Fox Software 386-MATLAB MathWorks High C Professional Pascal MetaWare MetaWare

• 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 •

C CODE FOR THE PC

source code, of course

C Source Code

Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	$400
CQL Query System (SQL retrievals plus windows)	$325
GraphiC 4.1 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	$325
Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	$300
Greenleaf Data Windows (windows, menus, data entry, interactive form design)	$295
Vitamin C (MacWindows)	$200
resident C (TSRify C programs, DOS shared libraries)	$165
Essential C Utility Library (400 useful C functions)	$160
Essential Communications Library (C functions for RS-232-based communication systems)	$160
Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	$150
Greenleaf Functions (296 useful C functions, all DOS services)	$150
OS/88 (U**x-like O/S, many tools, cross-development from MS-DOS)	$150
Turbo G Graphics Library (all popular adapters, hidden line removal)	$135
CBTree (B+tree ISAM driver, multiple variable-length keys)	$115
MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	$115
PC/IP (CMU/MIT TCP/IP implementation for PCs)	$100
B-Tree Library & ISAM Driver (file system utilities by Softfocus)	$100
The Profiler (program execution profile tool)	$100
Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	$100
Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	$100
Wendin O/S Construction Kit or PCNX, PCVMS O/S Shells	$95
QC88 C Compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	$90
JATE Async Terminal Emulator (includes file transfer and menu subsystem)	$80
MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	$80
ME (programmer's editor with C-like macro language by Magma Software)	$75
WKS Library (C program interface to Lotus 1-2-3 program & files)	$65
Quincy (interactive C interpreter)	$60
EZ_ASM (assembly language macros bridging C and MASM)	$60
PTree (parse tree management)	$60
HELP (pop-up help system builder)	$50
Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	$50
Heap Expander (dynamic memory manager for expanded memory)	$50
Make (macros, all languages, built-in rules)	$50
Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	$50
Coder's Prolog (inference engine for use with C programs)	$45
C-Help (pop-up help for C programmers ... add your own notes)	$40
Biggerstaff's System Tools (multi-tasking window manager kit)	$40
CLIPS (rule-based expert system generator, Version 4.0)	$35
TELE Kernel (Ken Berry's multi-tasking kernel)	$30
TELE Windows (Ken Berry's window package)	$30
Clisp (Lisp interpreter with extensive internals documentation)	$30
Translate Rules to C (YACC-like function generator for rule-based systems)	$30
6-Pack of Editors (six public domain editors for use, study & hacking)	$30
ICON (string and list processing language, Version 6 and update)	$25
LEX (lexical analyzer generator)	$25
Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	$25
AutoTrace (program tracer and memory trasher catcher)	$25
C Compiler Torture Test (checks a C compiler against K & R)	$20
Benchmark Package (C compiler, PC hardware, and Unix system)	$20
TN3270 (remote login to IBM VM/CMS as a 3270 terminal on a 3274 controller)	$20
A68 (68000 cross-assembler)	$20
List-Pac (C functions for lists, stacks, and queues)	$20
XLT Macro Processor (general purpose text translator)	$20
C Tools (exception macros, wc, pp, roff, grep, printf, hash, declare, banner, Pascal-to-C)	$15

Data

WordCruncher (text retrieval & document analysis program)	$275
DNA Sequences (GenBank 48.0 of 10,913 sequences with fast similarity search program)	$150
Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	$60
Webster's Second Dictionary (234,932 words)	$60
U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	$35
The World Digitized (100,000 longitude/latitude of world country boundaries)	$30
KST Fonts (13,200 characters in 139 mixed fonts: specify TEX or bitmap format)	$30
USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	$20
NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	$15
U. S. Map (15,701 points of state boundaries)	$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409 USA
acw!info@uunet.uu.net
Voice: (512) 258-0785
BBS: (512) 258-8831
FidoNet: 1:382/12

Free surface shipping on prepaid orders

MasterCard/VISA

by Michael Cherry

3270 Gateways

LAN workstation users sometimes need a connection to an IBM mainframe. When such a connection is required, one of the easiest ways to make a connection is with a 3270 gateway.

The IBM Mainframe Network Environment

In a typical IBM mainframe network environment, terminals are connected to a cluster control unit via coaxial cable. The cluster controller, in turn, is connected to a front-end processor attached to the host computer (Figure 1).

A confusing aspect of the IBM mainframe environment is that it uses a numeric naming scheme to identify the equipment. Table 1 identifies the common IBM equipment numbers, with descriptions.

Initially, a PC was added to an IBM network by installing a Digital Communications Associates (DCA) IRMA card in the PC. This permitted a coaxial connection between the PC and the cluster control unit (Figure 2) which allowed the PC to emulate a 3278 terminal. This was very workable where there was a cluster control unit to attach to, however, this is not always the case.

Therefore, in addition to the IRMA-type connection, other links were developed that enabled a PC to emulate both a 3278 workstation and a 3274 cluster control unit (Figure 3). This approach was unpopular because it tied up a large portion of a front-end processor for only one terminal.

The introduction of PC LANs created an environment where the LAN workstations could share a host connection. All that was needed was a LAN gateway that emulated the cluster controller (Figure 4).

There are two fundamental reasons for using a gateway: terminal emulation and file transfer.

Terminal Emulation

There are some differences between a PC emulating a 3278 or a 3279 terminal and the real thing. First, the PC's display is not the same as the display on a 3270 terminal. Messages displayed on status line 25 are different; for example, a 3278 terminal uses a "lightning bolt" symbol with a number beside it to indicate a communication check. The PC cannot readily produce the lightning bolt symbol and therefore other symbols must be used.

Second, the keyboards of the PC and the 3270 terminal are different. Several keys on a 3270 are critical to the execution of applications on the IBM mainframe. For example, many programs require Program Function keys to select items from menus or to invoke certain functions in editors or applications. The PC's 3270 emulation must remap the PC keyboard to provide these functions, which often leads to awkward keystrokes.

A third factor is how much memory the emulation software demands. If the emulation requires too much memory, there may be insufficient memory space to run an application when you "hot-key" between the 3270 terminal emulation and DOS.

File Transfer

Originally, programs to transfer files between IBM mainframes and PCs required starting an edit session on the mainframe and capturing the "listing" file on the PC during the edit session. As a result, you often got edit prompts in addition to the file data.

When IBM introduced its 3270PC, it combined a personal computer with a 3270 terminal. Included was a file transfer program that is often referred to as IND$FILE (the proper name is the IBM 3270 File Transfer Program). We like to use this file transfer software because the drivers for this program are generally installed on the IBM mainframe, and because they work well. The IRMA file transfer programs, FT/TSO and FT/CMS, are also widely distributed.

In addition, there are third-party file transfer programs, including one popular program, Tempus-Link. It makes files stored on the mainframe appear as if they were on a virtual disk drive on the PC. Using such pro-

Table 1. Common Equipment Numbers and Descriptions

Terminals and Printers	
3278 Model 2	1920 character, 24 PF keys
3279 S2A	1920 character, four colors
3279 S2B	1920 character, seven colors
3279 S3G	2560 character, seven colors, presentation graphics, extended functions and programmed symbols
3287 Model 2	120 characters per second
Control Units	
3174	New controller
3274 51C	For SDLC line control 32 devices
3274 61C	For SDLC line control 32 devices

grams, file transfer is accomplished by doing a file copy between the physical disk on the PC and the virtual disk on the mainframe.

LU Pooling

In the IBM network, a connection between a terminal and the host is referred to as a "Logical Unit" (LU). Each terminal requires one LU. Therefore, a cluster control unit set up to support 32 LUs can have up to 32 terminals connected to it using coaxial cable.

> Some application software uses LU assignments for security . . . a gateway should use pooled and assigned LUs.

When a gateway is in use, the LUs can be pooled. Our experience has shown that a 32-LU gateway can support approximately 50 workstations. The key is to make sure that no more than 32 users can be logged onto the host through the gateway at one time. When a user accesses the host through the gateway, an LU is assigned to the workstation. The next user to log on gets the next available LU. If the first user logs off, the LU he or she was using is returned to the pool and becomes available for the next user to request a host connection.

While this can be an advantage, it is sometimes necessary to assign specific LUs to specific workstations. Some application software use LU assignments for security. Therefore, a gateway should be capable of using pooled and assigned LUs. The gateways we use on the LANs we design include: NAS SBS, Eicon SDLC and x.25, INS, IRMA-

Figure 1. Typical IBM Network Environment

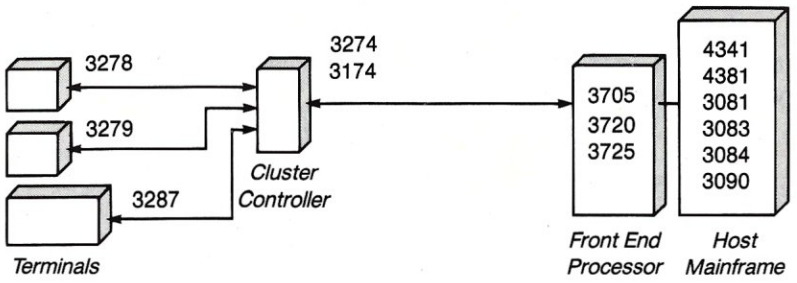


Figure 2. PC connected to IBM LAN via an IRMA card.

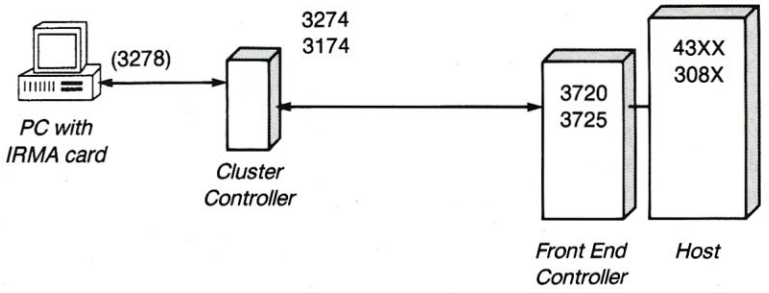


Figure 3. PC connected to IBM via 3278/3274 Emulator card.

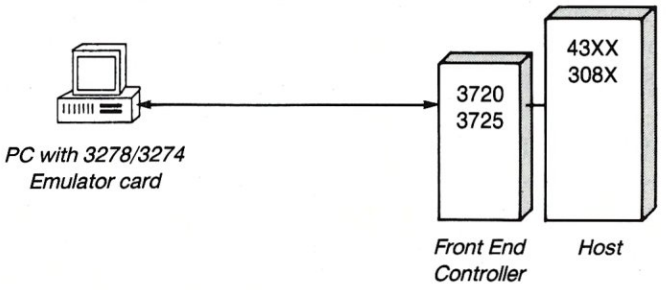
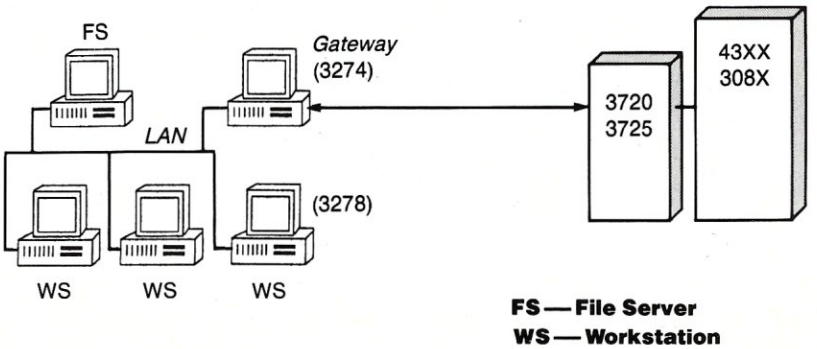


Figure 4. PC LAN connected to IBM system via a gateway.



LAN, and Novell/CXI.

We select gateways to meet specific customer requirements, based on an analysis of both the customer's host and LAN environment. The following sections offer a brief summary of the gateways and their salient features.

The NAS SBS Gateway

National Advanced Systems (NAS) offers a unique gateway product called the Single Board Solution (SBS). The SBS gateway supports the Novell IPX (Internetwork Packet Exchange) allowing users to access the gateway across a bridge. This cannot be done with NetBIOS gateways.

Further, software also is available that enables the SBS hardware to operate as a 3770 Remote Job Entry (RJE) gateway. While many users want terminal emulation, there is probably no faster file transfer facility than RJE.

The Eicon SDLC Gateway

Eicon offers an excellent gateway product that works well, and is perhaps the easiest gateway to install. Of all the available gateways, it provides the best 3279 3GS and IBM Application Programming Interface (API) emulation. The Eicon gateways also supports x.25 network connections.

Eicon's technical support is excellent. Recently, we found a file transfer problem and Eicon programmed a repair that was sent out the next day by Federal Express. Most gateway manufacturers take weeks to even acknowledge a problem.

The INS Gateway

INS offers gateways for IBM mainframes and for 8100 processors. The 8100 gateway is the only one we know of for 8100 processors. The INS gateways are moderately easy to install and function well.

The IRMA Gateways

Digital Communications Associates offers an IRMA gateway that is as good as its coaxial attach card. It is moderately easy to install and works quite well.

The Novell/CXI Gateway

The Novell/CXI gateway works, but it is the most difficult to install and lacks technical support. The documentation is obscure at best, and it can be very hard to find exactly where you should be in the installation process.

Technical support may improve in the future, since Novell recently acquired CXI. The difficulty in getting answers or product from CXI may be related to the new organizational problems encountered during this period of transition.

Table 2. An AUTOEXEC.BAT File to Load an Eicon Gateway

`ANET3`	Load Novell network shell
`NETBIOS`	Load NetBIOS
`NABIOS SRV IBMHOST /32`	Load NABIOS for gateway server
`SDLC START`	Start SDLC gateway component
`SNA START`	Start SNA gateway component

Table 3. The Workstation Program Needed for an Eicon Gateway

`ANET3.COM`	Novell workstation shell
`NETBIOS.COM`	Novell NetBIOS emulation program
`NABIOS.COM`	Eicon Network Adaptor BIOS
`SDLC.EXE`	Eicon SDLC program
`SNA.EXE`	Eicon SNA program

Installation Tips

While gateway manufacturers say that their gateways can be used in a non-dedicated workstation, we do not recommend this. We believe that the gateway should be installed in a dedicated workstation, and that the workstation should be located in a secured area.

With a non-dedicated workstation, a user might run an application that hangs the computer and requires rebooting the machine via the ALT + CTRL + DEL keys. This also reboots the gateway, which causes the active host sessions to be dropped. In addition, our experience is that when a PC containing the gateway is left in an accessible area, someone is going to come along, not realize that the PC is running the gateway, and reboot the system to run another application.

We believe that the gateway should be installed in a dedicated workstation, and that the workstation should be located in a secured area.

At HallComm NetWork Services, we have designed a special communications server to house 3270 gateways. It is diskless, and boots from a PROM installed on the network interface card. The communications server does not look like a PC, and therefore users leave it alone. Regardless of where the gateway is housed, it should be designed so the gateway is automatically initialized whenever the communications server is started. We accomplish this either through an AUTOEXEC.BAT file or by logging on a specialized

M BREAKOUT

Solve RS-232 compatibility problems. Fast.

A. *Pocket-Sized for easy transport*

B. *Current Loop Test for short haul modems 10, 20, 40 and 60mA*

C. *Open Circuit Voltage test for testing different ground potentials*

D. *Easy to read labels that can't wear off*

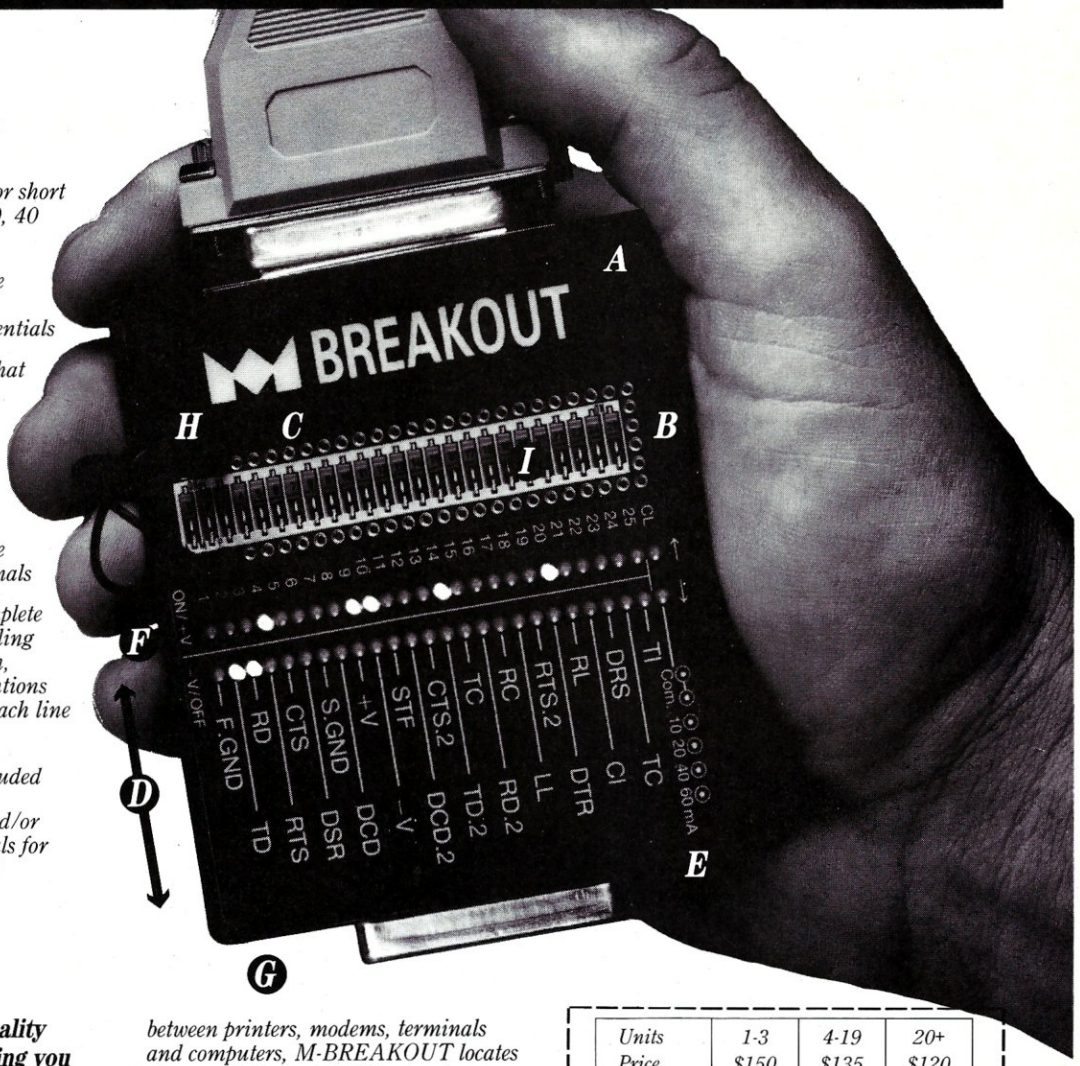
E. *High-impact ABS plastic case for years of dependable service*

F. *52 LED'S for 4-state indication on all signals*

G. *Reverse side has complete reference chart detailing DTE/DTC direction, CCITT/EIA conventions and descriptions of each line*

H. *Gold plated contacts. 8 Jumper cables included*

I. *Switches to break and/or re-direct all 25 signals for trouble-shooting and cable-making*



This pocket-sized, high quality breakout box has everything you need to make RS-232 trouble-shooting easy.

Made of high-impact ABS plastic, the M-Breakout tester is a must for engineers, in-house and field service technicians, and everyone who installs, demos, repairs or uses computer equipment. For problems with asynchronous serial interfaces, cables, hook-ups between printers, modems, terminals and computers, M-BREAKOUT locates the trouble fast. 52 LED's give 4 state (space/mark/clocking/none) indication on each signal. 25 in-line switches break and re-direct. Problem-solving and cable-making are easy. M-BREAKOUT uses bright, low-power LED's. That means the unit is always ready when you are. No batteries to worry about or replace.

Units	*1-3*	*4-19*	*20+*
Price	*$150*	*$135*	*$120*

☐ *Please send me ____ units @ $____ = ____*
California residents add 6.5% tax ____
UPS 2nd day air ($4 per unit outside CA) ____
if UPS COD add $3 ____
Enclosed is my check for Total $ ____

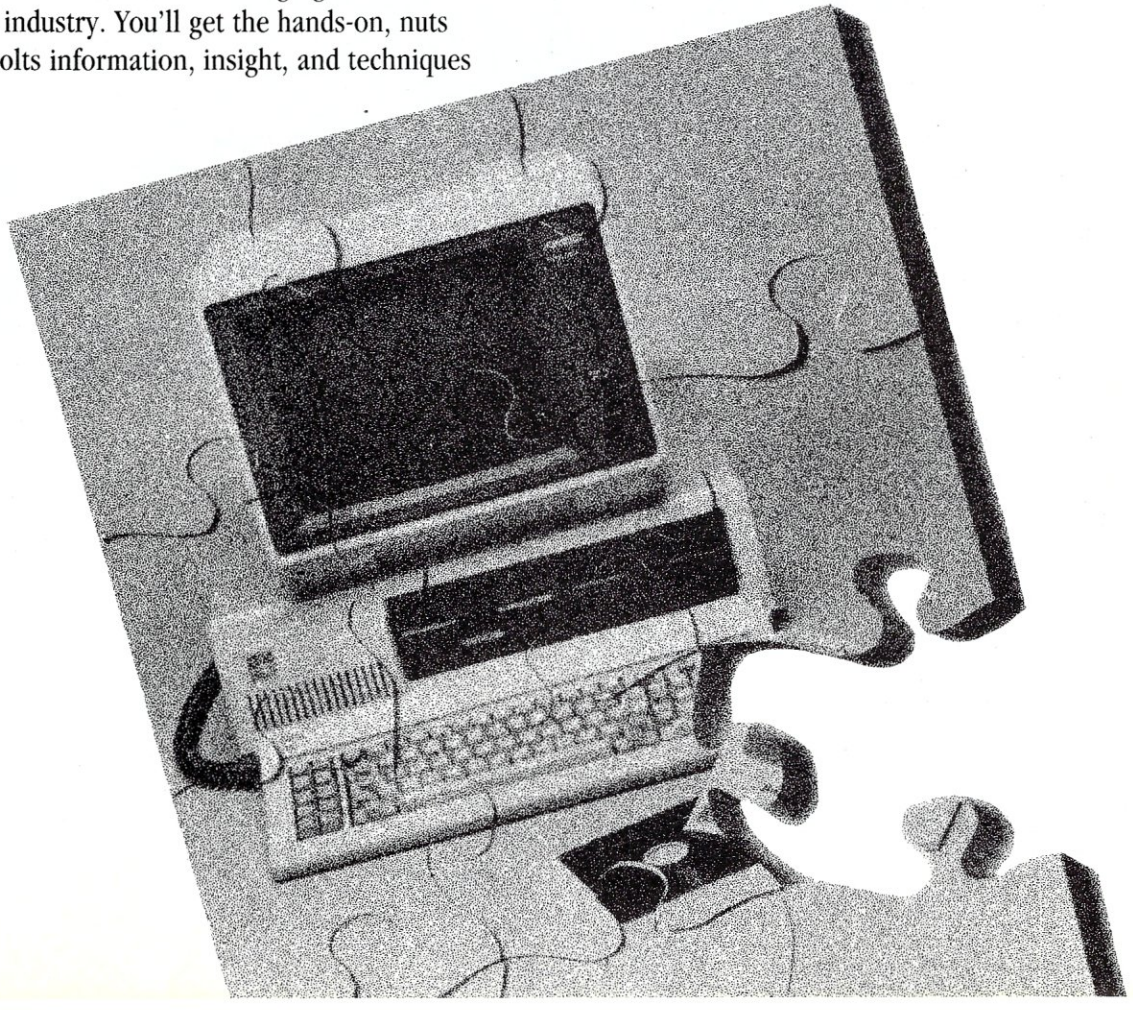
Name _____
Company _____
Address _____
City _____
State _____ *Zip* _____ - _____
Phone (_____) _____ - _____

☐ *Put me on your mail list*
30 Day Money-back Guarantee

M-Test Equipment
P.O. Box 146008, San Francisco, CA 94114-6008
415-861-2382 FAX 415-864-1076 TWX 710-111-4518

Why puzzle when you don't have to?

Micro/Systems Journal has the answers. Whether it's networking, systems integration, programming, or scientific computing questions, *M/SJ* will lead you out of the maze of microcomputer mayhem. With each issue you'll find comprehensive coverage of all the technical information that will keep you up-to-date with the ever-changing microcomputer industry. You'll get the hands-on, nuts and bolts information, insight, and techniques that *M/SJ* is famous for providing . . . in-depth tutorials, reviews, hints, the latest on multitasking, languages and operating systems. So stop your puzzling . . . subscribe right now and the answers will be yours. Simply drop the attached card in the mail—that's all there is to it.



Got Software Problems?

Get TURBO GhostWriter

Now $99.00

	Full Version $289	Starter Version $99	Cost of Upgrade for $99 version
B-Tree File Manager	Yes	Yes	N/A
Context-Sensitive Help	Yes		50
Command Window	Yes		50
Relational Model	Yes		150
Free Support	No Limit	1 Hour	100
Manual	Cloth D-ring	Paperback	50
All Upgrades	...	...	225

Orders & Information 800 227-7681

30 day money-back guarantee*(less $14 s/h)*

YES! Send me___copies of Turbo GhostWriter today.
Name _____
Address _____
City,State,Zip _____
Phone _____
MC, Visa, or Choice Card
#(or write COD) _____
Expiration Date _____
Circle version $99 $289

ASCII - 3239 Mill Run - Raleigh, NC 27612
800 227-7681

Turbo GhostWriter is an application generator that creates 80% of your custom application automatically -- as a Turbo Pascal Program. **That 80%** is done swiftly in just 10% of the time it takes to code it yourself.**That 80%** contains all of the "hooks" to add custom features such as table verification, security, currency conversion, importing files from other languages, etc...

Now it's up to you to write the code to relate one file to another. The relational model shows you how and it only takes a few minutes. End to end, all programming for 5 relational files should take less than 30 minutes. It's far superior to doing the whole thing from scratch. Your user sees a consistent interface, all programs function the same way, and your code is 100% error-free Turbo Pascal.

ASCII wants to give you a chance to try Turbo GhostWriter. For a limited time you can buy a fully functional version for just $99. This version will create flat files such as mailing lists, telephone files, inventory files, simple payroll files, etc. The utilities are the same as for the full-blown product, but are modestly packaged, and support is limited. It's the perfect way to see if an application generator is for you without risking a cent.

user who initializes the gateway and then logs off. The AUTOEXEC.BAT approach is easy, but requires that the files be stored where they can be accessed by the gateway (usually on the floppy drive of the gateway workstation or in the LOGIN directory). The advantage of the specialized login ID is that the gateway files can be accessed from the file server and you can see if the gateway is active by checking the status of the gateway user.

Table 2 is a sample of the AUTOEXEC.BAT file we use to load and start an Eicon gateway. When using this approach, the programs shown in Table 3 have to be available to the gateway workstation.

To achieve the same result, but log in a user called Eicon, the AUTOEXEC.BAT file might be:

```
ANET3
NETBIOS
F:
LOGIN EICON
```

The user, EICON, would then have the following statement in its individual user login script:

```
MAP G:=SYS:COMM/EICON
DRIVE G:
EXIT "START.BAT"
```

The G: drive would be the drive that was mapped to the directory with the Eicon software. START-.BAT could then be:

```
NABIOS SRV IBMHOST /32
SDLC START
SNA START
LOGOUT
```

Similar approches can be created for most manufacturers' gateways.

Another consideration is the hardware interrupt used by the LAN and gateway interface cards, both housed in the same communication server or workstation. Some LANs require the LAN interface cards to use interrupt 3, e.g., Ethernet with Novell, and some gateway cards are preset to interrupt 3 and are not alterable, e.g., INS. There are "patched" NetWare shells that use another interrupt. While this gets around the problem, many vendors will not or cannot support such patched shells.

Also, some file transfer software use the API (Application Program Interface). APIs can be different, and a file transfer written for one API may not work with another API. One popular API is the IBM API. This API uses software interrupt 7A (hex). Some technical reference manuals indicate that 7A is unassigned, however IBM uses 7A for the API and Novell uses 7A for the workstation shell. Thus, there can be a conflict in using the IBM API in a gateway on a Novell LAN.

Again, there are LAN shells that use interrupt 7B rather than 7A.

TASKVIEW

ROCK SOLID MULTITASKING!

Packed with the power you need, TASKVIEW takes you beyond the limits of DOS.

- Communicate while you edit
- Compile while you print
- Load up to 5 megabytes of programs
- Manage resident utilities
- Time-slice multiple jobs
- Cut & paste between programs

TASKVIEW lets you load up to 10 of your favorite applications and switch between them at a keystroke. They can even continue to run while you work on something else! TASKVIEW is the BEST multitasker you can buy. Just ask our customers.

"Thanks for a wonderful product!"

"TASKVIEW is the only multitasker I own which runs efficiently & correctly both on my Zenith Z-151... and on my Orchid 286e... I also have DoubleDOS, DESQview, TopView, Windows, and Concurrent PC-DOS".

"Fantastic! What DoubleDOS, MS-Windows, and all the other 'stuff' should have been!"

"My BBS is now up 24 hours, 7 days a week thanks to TASKVIEW."

"Thanks! I needed that!"

TASKVIEW requires an IBM PC, XT, AT or Jr compatible, and PC or MS DOS 2.x to 3.x. To get your copy, call toll free:

(800) 367-0651

or send $79.95 + $3.00 S&H ($8.00 Intl.) to:

Sunny Hill Software

PO Box 55278, Seattle, WA 98155-5278

For more information call (206) 367-0650

DoubleDOS trademark Softlogic Solutions, Concurrent PC-DOS reg. trademark Digital Research Corp., MS Windows trademark Microsoft Corp., DESQview trademark Quarterdeck Office Systems, Inc. TopView trademark IBM Corp.

This patch will allow you to run file transfer packages that use the IBM API, and can be used to connect 3270PCs to a network. However, the previous caution about patched shells applies here as well—you may not be able to get vendor support for a patched shell.

Conclusion

There are several gateways available that can be used to enable a PC user to access an IBM mainframe. No matter what you choose, points to consider in selecting a gateway include:

- What is the host application to be accessed?
- What terminals do you have to emulate?
- What file transfer has to be done?
- Does the gateway have to be accessed across a bridge?
- What speeds do you want to use?

The gateway that will work best for you will depend upon your requirements. □

Michael Cherry is Vice President of Technical Support for HallComm Network Services (HNS), a company devoted exclusively to designing and implementing LAN systems.

Product Information

Digital Communication Associates (DCA)
1000 Alderman Dr.
Alpharetta, GA 30201-4199
(404) 442-4000
(800) 241-IRMA

Eicon Technology
2196 32nd Ave.
Montreal, Canada H8T 3H7
(514) 631-2592

HallComm NetWork Services
8101 E. Prentice Ave., Ste. 304
Englewood, CO 80111
(303) 770-6387

INS
P.O. Box 91395
Mobile, AL 36691
(800) 762-3270

Micro Tempus Inc.
440 Dorchester Blvd., Ste. 300
Montreal, Canada H22 1V7
(514) 397-9512
(800) 361-4983

National Advanced Systems Inc.
9535A Waples St.
San Diego, CA 92121-2997
(619) 458-2000

Novell, Inc.
122 E. 1700 South
Provo, UT 84601
(800) 453-1267

If You Have Turbo C You Have Half Your C-Programming Vehicle

Turbo C is a great compiler but there is one vital cog missing—debugging. Without it, you have to spend an awful lot of energy to go a short distance.

Gimpel Software's C-terp, long recognized as the leading C interpreter, now fully supports Turbo C with complete compatibility guaranteed.

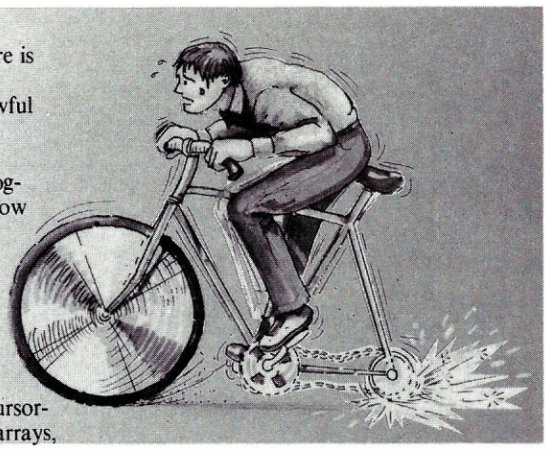
Interactive Debugger—Our debugging facilities include split screen (code in upper portion, dialog in lower), breakpoints (sticky, temporary, line/function, cursor-directed), display of structures and arrays, execution of any expression (even those involving macros), function traceback with arguments, watch expressions and watch conditions (watchpoints). Our watch expressions can be structs or arrays. We catch out-of-bounds pointers!

No Toy—Full K&R with ANSI enhancements. Multiple-module with a built-in automatic make. It has virtual memory option (with optional direct use of extended memory) and a shared symbol option for those big programs. It supports graphics, dual displays and the EGA 43-line mode.

Links to external libraries—(both code and data, automatically) which can call back to interpreted functions. Function pointers are compiler compatible.

100% Turbo-C compatible.—Same header (.h) files, data alignment, bit field orderings and preprocessor variables as your compiler. We link in your compiler's library.

Our reconfigurable editor—is multifile and comes with a configuration script to mimic Turbo's editor.



The missing wheel that will turn your half-cycle into a bicycle

C-terp

Order C-terp today!
Call (215) 584-4261

Introductory Price for Turbo C-terp:
$139.00
VISA, MC, COD—30 day money back guarantee

C-terp Version 3.0 is also available for the following compilers:
Microsoft, Lattice, Aztec, C86, and Mark Williams ($298) and Xenix ($498).

GIMPEL SOFTWARE
3207 Hogarth Lane
Collegeville, PA 19426

C-terp is a trademark of Gimpel Software, and Turbo C of Borland International.

by A. G. W. Cameron

Mini-supercomputers on the Desktop

Desktop number crunching will be revolutionized this spring. But before exploring why, let's start by defining some terms. The definition of "supercomputer" is changing gradually with time. Currently, it means a machine change with a computing power of 10 mflops (millions of floating point operations per second) or more—quite a lot more for the best known supercomputers from Cray. This benchmark is evaluated by seeing how fast the machine can carry out operations on a 100 × 100 matrix, which gives a reasonable mix of the most common floating point operations (the LINPACK benchmark). Usually, it also means a cost of at least $1,000,000.

There also has sprung up a new class of machine, the mini-supercomputer. Currently, this means a machine with a computing power in the range 1 to 10 mflops. The cost starts at $100,000 and escalates swiftly. This processing power is about to come to desktop micros in the form of the Weitek XL-Series coprocessor boards. But the cost will be just a shade over $10,000.

Desktop workstations have already been pressing at the bottom of this performance range. My Sun 3/260 workstation with floating point accelerator (based on the Weitek 1164 and 1165 chips) is rated at about 0.5 mflops. The new Compaq Deskpro 386/20 with the Weitek 1167 coprocessor board installed (see "The Scientific Computer User," *M/SJ*, February 1988) should have a performance of about 0.7 mflops. The new Sun 4/260 workstation is rated at 1.1 mflops and is based on the SPARC (Scalable Processor ARChitecture) technology, coupled with the Weitek 1164 and 1165 chips. This spring, the technology is scheduled to appear in the form of a coprocessor board from Definicon Systems (see "The Scientific Computer User," *M/SJ*, January 1988). Thus, by some technical reckoning, desktop machines will cross the mini-supercomputer threshold with the appearance of the Definicon board. But a much more powerful coprocessor board will appear at about the same time (it is already being sampled) that utilizes the XL-Series chips from Weitek.

The Weitek XL-Series

The XL-Series of boards are technically called "attached processors." The XL-8000 consists of an integer processing unit and a program sequencing unit. The XL-8032 adds to these a 32-bit floating point processing unit. The XL-8064 adds a 32-bit/64-bit floating point processing unit. The XL-8032 is available from Weitek as a two-board development unit for PC AT/Xenix machines and from Mercury Computer Systems as single, add-in boards for the PC AT operating under DOS, as well as for Sun and Apollo workstations and for DEC MicroVAXen. Weitek expects to have 8064 development units available in the first quarter of this year. Mercury has stated that its MC3200 (the XL-8032 in Mercury adaptation) is the first member of a family of upward-compatible products and that the addition of a 64-bit coprocessor is expected to come off the drawing boards in the not-too-distant future.

In addition to the integer and floating point processors, the XL-Series processors include a 32-element data register file and a 33-element program control stack. The floating point chips also have a 32-element floating point register file. These register files allow memory access to be reduced by maintaining variables and, where possible, passing parameters in registers rather than in RAM. The processors have a RISC (Reduced Instruction Set Computer) architecture, even though their instruction set is rather large. The integer instructions, except multiply and divide, complete in a single cycle. The floating point instructions, except divide, complete in no more than four cycles. The floating point units are pipelined so that a new operation can begin every cycle. The various functional units operate in parallel so that floating point, integer, memory, and control operations can all take place at the same time. Thus the XL-Series boards are designed to be vector processors. However, if they were only vector processors, they would form rather inflexible array processors, running as slave computers under a host. Actually, they can be run that way, but the element that makes them of great interest is that they are also designed to be efficient scalar pro-

cessors and can run as independent computers if desired, with the exception of the interface to the host that handles operating system services.

There are separate code and data buses on the XL-Series processors and each has a 32-bit address. The separation of the code from the data operations effectively doubles the bus bandwidth of the machine. The 8064 processor may be configured for either a 32-bit or a 64-bit data word, with corresponding data bus width. The latter allows double-precision floating point words to be loaded or stored in a single bus operation.

> The Parallelizer rearranges the instruction flow so that as many operations take place concurrently as possible.

Now let's consider floating point performance. The XL-8032 runs at 5 sustained mflops and 25 peak mflops (which gives some idea of the spread between performance in scalar operations and in very efficiently coded vector operations). The XL-8064 runs at 6 sustained mflops and at 20 peak mflops. These peak numbers are ideal values that will almost never be attained in real problems. Tony Funari, an engineer at Weitek, has told me that both the XL-8032 and the XL-8064 have run single-precision LINPACK at 8 mflops and the XL-8064 has run double-precision LINPACK at 13 mflops. The latter performance was achieved using a version of the XL-8064 with a 64-bit data bus; the version of the XL-8064 with a 32-bit data bus, which is essentially the 8032 board with an 8064 chip popped in instead of the 8032, runs the double-precision LINPACK at 12 mflops. Notice that there is only a modest gain to be had in going to the 64-bit data bus.

The software provided by Weitek includes C, Pascal, and FORTRAN compilers from Green Hills Software developed to run in UNIX systems. After passing through a compiler, a program passes through a Parallelizer which optimizes the code for the pipeline features of the processor board, and then it passes through a linker and librarian to produce an executable program. There also is an assembler that can be used to write tightly-coded routines for linking with the output of the higher level languages. Mercury has taken over the C and FORTRAN compilers from Weitek and has rewritten the board interface software so the board can run under DOS or under the operating systems of the various workstations mentioned above.

The Parallelizer is a particularly interesting development. It rearranges the instruction flow so that as many operations take place concurrently as possible. Thus, integer, floating point, and address generation operations can be scheduled to occur simultaneously. The Parallelizer can increase execution speed by up to a factor of two. There are three fields in the instruction word: the sequencer field (8 bits wide), the integer field (24 bits wide), and the floating point field (32 bits wide). Thus, every instruction word can provide operation codes for three parallel operations. A minority of the instructions use more than one of these fields and thus interfere with complete parallelism.

The prices charged by Mercury for its MC 3200 board and related software are $8,000 for the basic AT coprocessor board, including 2 megabytes of memory on board; $1,995 for the Driver/Executive, Linker, Scientific Algorithm Library, and binary run-time license; and $1,500 for a C or FORTRAN compiler. Additional memory is added in the form of a daughter board and costs about $1,000 per megabyte. The daughter board can add 2, 4, or 8 megabytes to the AT version of the board, and up to 16 megabytes to the version for the Sun workstation. The Sun version of the board costs

"Dear Genifer..."

Q: "dBASE has me stuck at the dot prompt and I'm coming unglued! Can you save me?

Help! I've been using dBASE for months at the dot prompt, but now I've got to write a stand-alone, end-user application. When I try to program, I get "Mismatched DO WHILE and ENDDO" and "Improper data type in subsubtotal expression." I just can't take it any more! Help me Genifer or I swear, I'll jump off the manual!"

- On The Edge

A: Dear On The Edge:

Take a deep breath and listen. All you need is Genifer - the application generator that creates custom-made applications in minutes. That means you don't have to know the details of dBASE programming or waste countless hours cranking out code! You see, *Genifer* quickly delivers clean, self-documented code that makes life worth living! No wonder Infoworld said, "Genifer creates programs that are more clearly written than most of the code we've seen produced by human dBASEIII programmers." If you're still in distress, go see your dealer right now. As dBASE users around the world know, Genifer cares!

To find out more about how Genifer can change *your* life, visit your dealer or call toll-free: **1-800-631-2229** for a FREE brochure.
In CA: 1-800-541-3366

Genifer.

For dBASE III applications without tears.

bytelcorporation

1029 Solano Avenue, Berkeley, CA 94706
(415) 527-1157 Telex: 176609

Trademark/Owner: Genifer/Bytel. dBase III/Ashton-Tate.
©Copyright 1987 Bytel Corporation.

goodbye dBase!

dBASE Programmers

You need it!
You can handle it!
dB2c is here now!

dB2c Offers:
- Version 2.0 complete with Translator and File Handlers.
- Extensive implementation of dBASE III+ with over 200 functions and commands in C source code.
- Contains our own File Handlers plus interfaces for Lattice's dBC and Faircom's c-tree.
- Supports screen I/O with ANSI.SYS or fast assembler routines.
- Support for Microsoft, Lattice and Turbo C compilers.
- Tutor features of translation combined with familiar syntax of the library eases the transition to 'C'.
- One version supports MS-DOS, Xenix, Unix, OS-9 and Concurrent DOS.

are you ready?

dB→c

Toolkit $299

DAVID J. MARSH

Call or Write:
SOFTWARE CONNECTION, INC.
POB 712, Ely, MN 55731
(218) 365-5097

about $2,000 more than the above prices. Presumably, the anticipated MC 6400 to come from Mercury will cost a little more than the MC 3200, but probably not much more.

Considering that one can approach supercomputer speeds for these prices, these coprocessor boards will be immensely more cost-effective than either real supercomputers or mini-supercomputers. The number-crunching revolution will get a big boost to a performance level for desktop machines that, just a few weeks ago, I did not think would be reached until the early 1990s.

Weitek manufactures several other numeric coprocessing chips that offer the potential for other fast machines a little later on. Particularly interesting are the 2264 multiplier/divider chip and the 2265 adder chip. These are intended to be used in efficiently pipelined operations; the chip set can maintain 12.5 mflops when pipelined for 64-bit multiplications.

The BIT Chips

From the above descriptions, the reader may get the impression that the future of number-crunching lies in the hands of Weitek. It just so happens that a very fast, competitive, numeric coprocessor chip set also is available now which was developed and is manufactured by Bipolar Integrated Technology Inc. (BIT) of Beaverton, Oregon. The chips use bipolar emitter-coupled logic (ECL) and have both ECL and TTL (transistor-transistor logic) I/O interfaces. The latter devices are slightly slower and cheaper, so I shall only report on the faster, all-ECL versions.

BIT has made a technical breakthrough by reducing the cell size of the chip elements from the dimensions needed in previous ECL chips. This has increased the already very fast speed of the chip and decreased power consumption while allowing denser packing of the elements.

In the Weitek chip sets discussed above, the attainment of a high floating point processing rate depends on efficient pipelining of the operations. In contrast, the BIT chips feature a flowthrough architecture without pipelining. That means that the speeds given below are available for scalar operations and there is no advantage to be obtained in vectorizing the code.

The 32-bit chip set consists of the B3110 multiplier and the B3120 arithmetic logic unit and sells for $640 each in lots of 100. Normally, these chips would be operated in conjunction with the B3210 five-port register file chip, which is organized as 64 18-bit words. The B3110 multiplier performs single (32-bit) and double (64-bit) precision multiplies

> These coprocessor boards will be more cost-effective than either real supercomputers or mini-supercomputers.

at 28 and 22 mflops, respectively (35 and 27 mflops for register-to-register operations). The B3120 ALU (arithmetic-logic unit) performs single- and double-precision adds and subtracts at 40 mflops (53 mflops for register-to-register operations), and performs both 32-bit and 64-bit integer operations at 100 million instructions per second (mips).

The chips conform to IEEE standards for arithmetic. The multiplier chip also does double-precision divides and square roots somewhat more slowly than multiplies (180 and 325 nanoseconds, respectively, versus 45 nanoseconds for the multiply). Other IEEE operations, such as exponentiation, logarithms, and trigonometric functions, would have to be flagged as exceptions and handed off to another coprocessor, just as the Weitek chips now hand these operations off to a 68881 in the Sun workstations. Otherwise, they would have to be handled in firmware.

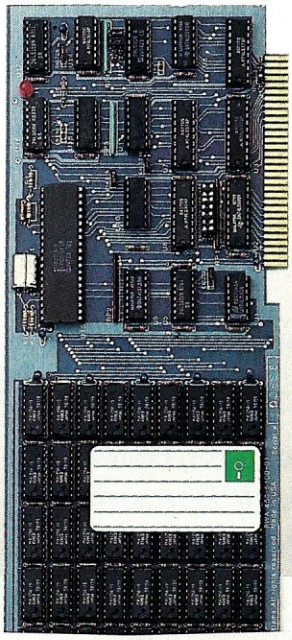
... continued on page 64

Think fast! Pick the better fit...

Our 5th Year Bonus! Mention this ad when ordering and get your choice of a V 20-8 (replaces 8088) or $20 off on your Battery Backup purchase!

FLOPPY DISK.

- Fills time between coffee breaks
- Makes a hard disk seem *fast*
- Your computer appears busy (even if you aren't!)
- Wears out moving parts



SEMIDISK Disk Emulator.

- Gets that job done NOW
- Makes a hard disk seem *slow*
- Maximizes your productivity with anything from databases to compilers
- Totally silent operation

...for YOUR demanding tasks.

SURPRISE! *Neither* is memory mapped, so they don't affect your precious Main Memory. *Both* retain data indefinitely - even with the computer turned off.

THE SEMIDISK SOLUTION. You could invest in a series of "upgrades" that turn out to be expensive band-aids without solving your real problem. Even those "Accelerator" and "Turbo" boards do little to speed up disk-bound computers. If your applications spend too much time reading and writing to disk (and whose don't?), you won't want to settle for anything less than a SemiDisk disk emulator. The SemiDisk comes in 512K and 2Mb capacity. More boards may be added to make up to an 8 Megabyte SemiDrive!

SPEED THAT'S COMPATIBLE. PC, XT or AT, if you need speed, the SemiDisk has it. How fast? Recent benchmarks show the SemiDisk is from 2 to 5 times faster than hard disks, and from 25% faster (writing) to several times faster (random reads) than VDISK and other RAMdisk software that gobble up your main memory.

MEMORY THAT'S STORAGE. Using our small external power supply, with battery backup, your data remains intact through your longest vacation or even a seven-hour power failure!

CELEBRATE WITH US! Now, SemiDisk celebrates its fifth birthday with a special offer for IBM-PC owners. Buy a SemiDisk now and we'll include an 8 MHz V-20 microprocessor (replaces the 8088) to make your new SemiDisk run even faster. Don't need the V-20? We'll take $20 off the price of your Battery Backup Unit!

	512K	2Mbyte
IBM, PC, XT, AT	**$495**	**$ 795**
Epson QX-10	**$495**	**$ 995**
S-100 SemiDisk II	**$795**	**$1295**
S-100 SemiDisk I	**$299**	**----------**
TRS-80 II, 12, 16	**$495**	**$ 995**
Battery Backup	**$130**	**$ 130**

Someday you'll get a SemiDisk.
Until then, you'll just have to...wait.

SemiDisk

MasterCard VISA

SemiDisk Systems, Inc., 11080 S.W. Allen Blvd., Beaverton, Oregon 97005 (503) 626-3104

S-100 Products

Turbo Slave I	$395 $410
Earth Net-100	$395 $410

PC Products

Earth Net PC	$375 $395

Prices & Shipping
& Handling CIA COD

System Manager
List **$375**

Complete TurboDOS Network Security Package

- All Menu Driven
- Restrict Drives and User Areas
- Log ON/OFF Record Keeping
- Full Electronic Mail
- Display System Status
- Talk-Send Messages to Another Processor
- Runs on All TurboDOS Systems

We Support Large TurboDOS Systems With Multiple Masters
We Support the Following OEM's
Earth ICM ICD L/F Teletek
Dealer Discounts

Multi-User Networking Specialists

Lodden Technology Limited

2834 N. Milford Rd.
Milford, MI 48042
(313) 685-3877

TurboDOS is a registered trade mark of Software 2000, Inc.
System Manager © Lodden Technology LTD

Continued from page 62

I would love to have a coprocessor board that uses the BIT chips. Unfortunately, that is for some time in the future. It is a somewhat daunting task to go from the 25-MHz coprocessor boards available today to the 100-MHz boards that will run the BIT chips. The interface to a relatively slow bus, such as the AT bus or the VME bus for Sun workstations, will be a tricky one. One wonders if some sort of buffering hardware would improve this interface. Then there is the problem of getting the appropriate software implemented for the system—the fact that it will be a scalar processor may simplify this. For the moment, we can only note that the hardware basis for a true supercomputer on a desktop now exists.

Obviously, a particularly critical element will be the on-board RAM. It is generally true that, for scientific and technical work, the faster the processor, the larger the memory size required. Yet when one needs very small cycle times and memory random access times, it tends to be the case that more expensive fabrication techniques are required using more costly materials, and the resulting memory capacities of the chips are reduced. The combination of larger memory needs and smaller chip capacities translates into high costs. Obviously, a very efficient caching arrangement will be important.

One cannot help wondering whether the techniques that have enabled such a fast scalar machine to be built might not be applied to build a vector machine with pipelining. A 45-nanosecond, double-precision multiply clearly takes about 5 clock cycles at 100 MHz. That is comparable to the situation in the Weitek XL-Series chips, where a new multiply can start nearly every clock cycle. A peak vector performance approaching 200 mflops might thus result.

It is extremely exciting to think that machines with this level of performance are likely to be available some time in the 1990s. They will certainly transform scientific and technical number-crunching, and I look forward to personal benefits from that. But many other technical benefits will also occur. Consider the effect on graphics, for example. One should be able to generate computer movies in real time (but only by upgrading display systems to use much larger bandwidths). Scientific computations are increasingly relying on sophisticated displays to enable the programmer to understand what is going on in a complicated calculation.

Thus, even though the scientific number cruncher always feels inadequately supplied with what he calls "REAL*8" operations, the time is swiftly approaching when the performance of all the other parts of the system must be upgraded along with the mflops performance. □

A. G. W. Cameron is Professor of Astronomy at the Harvard-Smithsonian Center for Astrophysics.

Product Information

Bipolar Integrated Technology Inc.
1050 N. W. Compton Dr.
Beaverton, OR 97006
(503) 629-5490

Definicon Systems Inc.
1100 Business Center Circle
Newberry Park, CA 91320
(805) 499-0652

Green Hills Software Inc.
425 E. Colorado St.
Glendale, CA 91205
(818) 246-5555

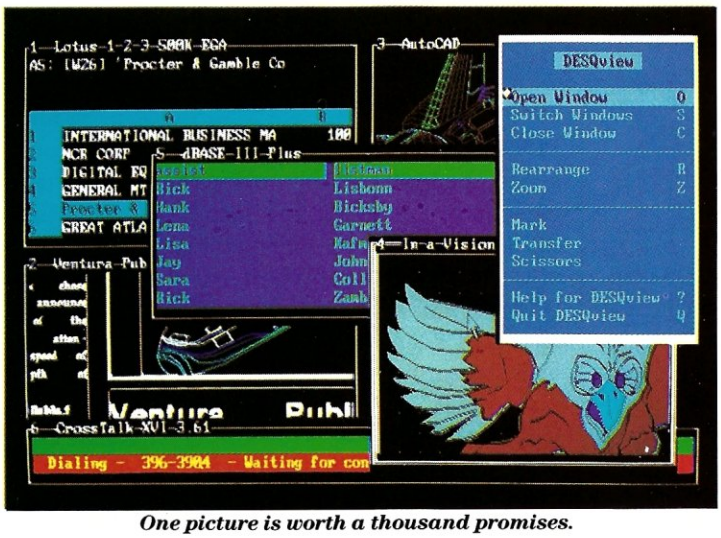
Mercury Computer Systems, Inc.
600 Suffolk St.
Lowell, MA 01854
(617) 458-3100

Sun Microsystems Inc.
2550 Garcia Ave.
Mountain View, CA 94043
(415) 960-1300

Weitek Corporation
1060 E. Arques Ave.
Sunnyvale, CA 94086
(408) 738-8400

How to tell the difference between DESQview™ 2.0 and any other environment.

Selecting DESQview, the environment of choice, can give you the productivity and power you crave, without the loss of your old programs and hardware. If you like your existing programs, want to use them together, transfer data between them, print, sort, communicate with or process-in-background, yet still have the need to keep in place your favorite PC(8088, 8086, 80286 or 80386), DESQview is the "proven true" multitasking, multi-windowing environment for you. Best of all, DESQview 2.0 is here now, with all the money saving, time saving, and productivity features that others can only promise for the all-too-distant future.



One picture is worth a thousand promises.

And with DESQview's new graphics enhancements for Hercules, CGA, EGA, and VGA, Version 2.0 still offers the same award winning and pioneering features for programs that earned DESQview its leadership, only now you can also run desktop publishing programs, CAD programs, even GEM-™, Topview-™, and Microsoft Windows-™ specific programs. In some cases you'll add as little as 10–40K to your system overhead. Now you can have multi-tasking, multi-windowing, break the 640K habit too and still get an auto dialer, macros, menus for DOS and, for advanced users, a new complete application programmer's interface capability. No wonder that over the years, and especially in recent months, DESQview, and now DESQview 2.0 have earned extravagant praise from some of the most respected magazines in the industry.

"Product of the Year" by readers vote in InfoWorld.

"Best PC Environment" by popular vote at Comdex Fall in PC Tech Journal's "System Builder" Contest.

"—I wouldn't want to run an IBM or compatible computer without DESQview"—InfoWorld, Michael Miller.

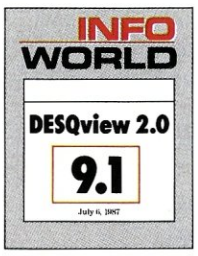
"A colossus among windowing environments"... "will run almost anything"—PC Week, Marvin Bryan.

"Windows, promises, but DESQview delivers"—MICROTIMES, Birell Walsh.

No other environment has consistently pioneered features, openness, and productivity. See for yourself. Send in the coupon. The possibilities are endless with DESQview 2.0.

Attention Programmers: For more information about Quarterdeck's API, and future 386 program extensions, call us today.

SYSTEM REQUIREMENTS
IBM Personal Computer and 100% compatibles (with 8086, 8088, 80286 or 80386 processors) with monochrome or color display; IBM Personal System/2• Memory: 640K recommended; for DESQview itself 0-145K• Expanded Memory (Optional): expanded memory boards compatible with the Intel AboveBoard; enhanced expanded memory boards compatible with the AST RAMpage• Disk: Two diskette drives or one diskette drive and a hard disk• Graphics Card (Optional): Hercules, IBM Color/Graphics (CGA), IBM Enhanced Graphics (EGA), IBM Personal System/2 Advanced Graphics (VGA)• Mouse (Optional): Mouse Systems, Microsoft and compatibles• Modem for Auto-Dialer (Optional): Hayes or Compatible• Operating System: PC-DOS 2.0-3.3; MS-DOS2.0-3.2• Software: Most PC-DOS and MS-DOS application programs; programs specific to TopView 1.1, GEM 1.1 and Microsoft Windows 1.03• Media: DESQview 2.0 is available on either 5¼" or 3½" floppy diskettes



Rush me DESQview 2.0! Today!

No. of Copies	Media 3½"/5¼"	Product	Retail Price ea.	Total
		DESQview 2.0	$129.95	$
		Shipping & Handling USA	$ 5.00	$
		Outside USA	$ 10.00	
		Sales Tax (CA residents)	6.5%	$
			Amount Enclosed	$

Payment: ☐ Visa ☐ MC ☐ AMEX ☐ Check

Credit Card: Valid Since ____/____ Expiration ____/____

Card Number:

Credit Card Name ____

Shipping Address ____

City ____ State ____ Zip ____ Telephone ____

Mail to: Quarterdeck Office Systems, 150 Pico Boulevard, Santa Monica, CA 90405.
NOTE: If you own DESQview call us for a special upgrade offer, or send in your DESQview registration card. AST Special Edition users included.

DESQview™
Quarterdeck

Quarterdeck Office Systems • 150 Pico Boulevard, Santa Monica, CA 90405 • (213) 392-9851

DESQview is a trademark of Quarterdeck Office Systems. AboveBoard is a trademark of Intel Corporation. Hayes is a trademark of Hayes MicroComputer Products Inc. IBM, PC, Personal System/2 and TopView are trademarks of International Business Machines Corporation. Microsoft Windows and MS are registered trademarks of Microsoft Corporation. Mouse Systems is a trademark of Metagraphics/Mouse Systems. RAMpage is a trademark of AST Research, Inc. GEM is a trademark of Digital Research. Hercules is a trademark of Hercules.

Charles H. Strom

A DOS Enhancement and Powerful Printer Utility

Another year of DOS-compatible public domain and shareware software is upon us. And the first software we'll look at is Norm Patriquin's PPRINT, version 3.1, current as of this writing.

PPRINT

PPRINT is user-customizable so it can be used with virtually any printer. Although a number of printer definition files are included in the shareware distribution, full instructions are provided for constructing PPRINT.DEF files. Support of the Hewlett-Packard LaserJet is an unusual plus. Some of the commands supported include font control, multiple copies, variable spacing, and a wildcard facility with printing based on file dates (for example, only those files that have date stamps newer than five days ago are printed). Also included are headers and footers, special formatting for C-source programs, multiple-space compression, variable margins, and so on. The list of options is a little overwhelming, but as Patriquin points out, batch files can be used conveniently since everything is command-line driven.

Norm Patriquin distributes PPRINT through the shareware principle—try it for free and register the program for $15 (registration only) or $30 (registration, disk and manual) if it satisfies your needs. PPRINT is but one of Norm's creations, and I will probably be discussing others in the coming months. PPRINT and other Patriquin utilities are available through his own BBS at (714) 369-9766 (modem), on GEnie's IBM RoundTable, and on many bulletin board systems.

Command Plus

Have you ever wondered why DOS's COMMAND.COM can be replaced with an alternative command shell? I finally found the answer in Command Plus, a useful (and as far as I can determine, the only) alternative to COMMAND.COM. Command Plus, by ESP Software Systems (11965 Venice Blvd., Ste. 309, Los Angeles, CA 90006) is not public domain or shareware, but it is low in cost. It retails for $79.95 and is available for a discount from The Programmer's Connection. Call them at (800) 336-1166.

Command Plus is a complete replacement for COMMAND.COM. No learning is necessary, as all the familiar DOS commands are used. It can be run under Versions 2.x or 3.x of DOS by simply invoking it as a TSR (terminate-and-stay-resident program) or under 3.x with the Shell command to completely replace COMMAND.COM.

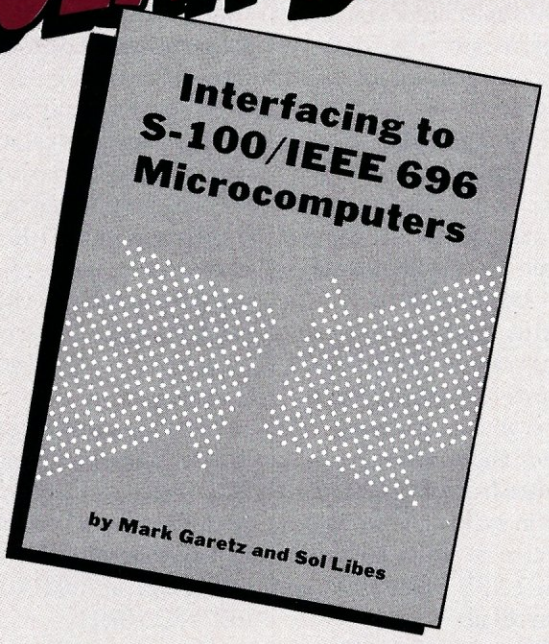
Command Plus's enhancements over standard DOS include several features for which I previously used add-ons discussed in earlier columns, such as CED and LIST. For example, there is a command buffer scrolled through with the cursor up and down arrows. Alternatively, a History command presents the command buffer in a numbered list. There is a full command-line-editing capability as well. Several of the familiar commands are greatly enhanced. DIR, for example, allows sorting by date, time, name, size, and extension. The long form of the command displays file attributes; hidden files can be displayed; and so on. The Copy command has expanded selection criteria, such as date/time ranges, verification before copy, and others. Similarly, the DELete function is more useful than its DOS ancestor.

Command Plus includes many other capabilities as well. An Alias command can generate RAM-resident batch files, modify the default options of any command, and so on. Browse is similar in usage to Vern Buerg's List, although it is not as rich in options. A Log facility allows automatic or manual tracking of machine usage for such purposes as billing, IRS certification, and others. Finally, ESP includes Script, an interpretive command processor similar in concept to Batch but much more powerful. Script files may call other scripts, but are limited to 10 levels of nesting. Each script file can access up to five each of integer, long integer, and string variables. Operators include comparisons (less than, equal, greater than or equal to, and so on), Booleans (NOT, AND, OR), arithmetics, conditionals, and others. BEGIN . . . END and CASE . . . ENDCASE constructs, FOR loops, IF . . . ELSE, and so on are also included. It is not necessary to go into further detail to see that SCRIPT offers great functionality to those willing to tap its power.

That's it for this month. Keep those cards and letters coming, and please upload your favorite public domain and shareware programs to the GEnie IBM RoundTable so I can get a chance to take a look at them! □

Charles Strom is a chemist-in-training who has been working with microcomputers for more than 10 years. He is a sysop on the GEnie national time-sharing service.

BACK BY POPULAR DEMAND



Interfacing to S-100/IEEE 696 Microcomputers by Mark Garetz and Sol Libes

Interfacing to S-100/IEEE 696 Microcomputers provides an in-depth look at how the S-100 bus works, and includes concepts that are basic to the understanding of most any bus-based system. You'll find:

- a precise description of the mechanical and functional design of the S-100 bus
- logical and electrical relationships
- bus interconnections and busing techniques
- descriptions of both parallel and serial interfacing as well as interfacing to RAM, ROM, and the real world
- a discussion of digital-to-analog and analog-to-digital conversion
- interrupts
- programmable timer/counters
- temporary master access and temporary bus masters
- and useful circuits.

While the examples contained in ***Interfacing to S-100/IEEE 696 Microcomputers*** relate specifically to the S-100 bus, the concepts presented can help you to expand the utility and power of any bus based system.

To Order: Return this coupon with your payment to M&T Publishing, 501 Galveston Drive, Redwood City, CA 94063.
Or, **CALL TOLL-FREE 800-533-4372** (in CA 800-356-2002)

Yes! Send me ***Interfacing to S-100/IEEE 696 Microcomputers*** $24.95

CA residents add sales tax _____ % _____

Shipping $2.25

TOTAL _____

☐ Check enclosed. Make payable to M&T Books.

Charge my ☐ VISA ☐ M/C ☐ Amer. Exp.

Card # _____ Exp. Date _____

Name _____

Address _____

City _____ State _____ Zip _____

40B

New Products

Manufacturers who would like to have their products listed here should send their news releases to the Managing Editor, Micro/Systems Journal, *501 Galveston Drive, Redwood City, CA 94063.*

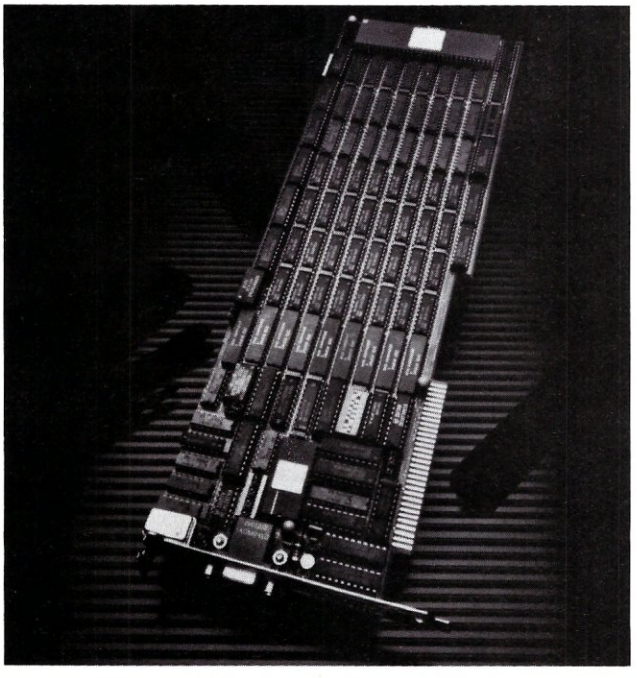
PC-Compatible Products

Special Offer for AutoCAD Board

Control Systems, Inc., has introduced the Artist 10, a high-performance graphics controller based on the Hitachi CRT controller IC to enhance AutoCAD version 9.0 on PCs. The board offers 1024 × 768 non-interlaced or 1024 × 1024 interlaced resolution with 16 or 256 displayable colors from a palette of 262,144 colors. The board is compatible with the IBM PC, XT, and AT and operates in concert with Control Systems' Artist GT, a software interlace option that works with system EMS memory to speed redraws and provide bird's-eye-view windows and multiple views.

To promote the Artist 10, Control Systems will offer an upgrade program through March 31. Targeted for customers using the Artist family of boards, qualified participants can return their old boards to Control Systems with a cashier's check and an upgrade certificate from their dealer and in exchange they will receive an Artist 10 board for $1,995, $1,000 less than retail.

For more information, contact Control System, 2675 Patton Rd., St. Paul, MN 55113; (612) 631-7800.



The Artist 10 from Control Systems, Inc.

386 Accelerator Now Available for PC and XT

Applied Reasoning Corporation has begun delivery of a 386 plug-in accelerator board for IBM or compatible computers. The PC-Elevator 386 comes with disk cache, RAM disk, and EMS support and runs an Intel 80386 processor at 16 MHz (no wait states) with a 32-bit bus. The board includes one megabyte of RAM operating at 100 nanoseconds, and is expandable to 16 MB with daughter cards. The company rates the board at 3.4 mips.

The PC-Elevator 386 lists for $1,995. For more information, contact Applied Reasoning Corporation, 86 Sherman St., Cambridge, MA 02140; (617) 492-0700

Other Hardware Products

Astrocom Gets AT&T Approval for Multiplexer

Astrocom Corporation has unveiled its 8000 Subrate Data Multiplexer, which is compatible with AT&T's ABC service. The SDM Series 8000 is a multiple-port, synchronous, time division multiplexer with an internal DSU/CSU. Modular design allows the user to customize the hardware required to create a specific application configuration providing capability and flexibility beyond that outlined in publication 54075. The multiplexer has modular expansion for 5-10-15-20 primary user ports and synchronous port speeds of 2.4 to 19.2K bps. It also has asynchronous secondary channels for network management and diagnostics; front-panel LCD command screen and control switches for data rate selection, configuration, and diagnostics; and a supervisory port to provide end-to-end network management functions such as network status, test initiation, and diagnostics.

The five-port unit sells for $3,350. For more information about the SDM Series 8000, contact Astrocom Corporation, 120 W. Plato Blvd., St. Paul, MN 55107; (612) 227-8651.

MSI Adds Two Multiplexers

Method Systems Inc. (MSI) has added two new multiplexers to its data communications line, the Mini Mux and the Mighty Mux. Both multiplexers have internal jumpering for easy modification providing options for transmission distances of one, two, and three miles (with some bit rate loss at longer distances). The Mini Mux operates at data rates up to a maximum of 9.9 kbits/sec., with receive/transmit lines for each of it eight channels. The Mighty Mux offers independent data rates of up to 119.2 kbits/sec. per channel, and each channel has a receive, transmit, CTS/RTS, or DTR/DCD line accommodating hardware handshake. Both units

SELF-INKING PRINTER RIBBON

Awarded United States Patent #4701062
Lasts 10-15 times longer than the conventional ribbon.
For printers using 1/2" width open spool ribbon:

- Okidata-82A-83A-84-92-93
- Teletype-33, 35
- Star Gemini 10X
- Extel
- Dec LA 180/120
- Dec LA 30/IBM 1443
- Teletype-Model 40
- Texas Instrument 800/810, 820, 880

CONTROLLED PRINTOUT DEVICES, INC.
P.O. Box 869, Baldwin Road
Arden, NC 28704
(704) 684-9044 • Telex: (Filmon-Aren) **577454**
Contact us by mail, phone or telex and we will forward you a brochure

A Reliable PC/XT Compatible for the Cornerstone of Your Products

NEW! Announcing The **SLY40-XT** NEW!

- Small Board Size (4-1/4" By 9-1/4"), Four Layer
- High Integration — Composed of just 17 Low Power CMOS ICS
- NEC's 8 MHZ V40
- One Megabyte of Zero Wait State RAM
- Ideal For Tough Industrial, OEM and Portable Applications
- American Made and Fully Supported by Slicer

PC and XT Are Trademarks of International Business Machines

MasterCard
Visa
Check
Money Order
C.O.D.

SLICER™

Slicer Computers Inc.
3450 Snelling Ave. So.
Minneapolis, MN 55406
612/724-2710
Telex 501357
SLICER UD

The Custom 386 Programmer's Workstation

"The winner, though, was the CAE/SAR 386. Its ESDI hard disk interface made it the fastest of all the machines in the disk access test."
PC Magazine
Dec. 22, 1987

Looking for a lightning-quick 386 system that's tailored to your needs? CAE/SAR Systems, Inc. will custom-fit you a 386 system more powerful than most on the market. Whether it's a system designed for your program development, artificial intelligence, CAE, or systems design work, CAE/SAR delivers reliable, powerful 386 workstations built for today's programmers.

Based on a proven 386 motherboard, CAE/SAR 386 systems come in dozens of different configurations for memory, disks, floating point and graphics. You can select high speed drives (16 ms), 70Mb, 140Mb, or 300Mb; EGA or mono monitors and cards; and 2.5Mb, 4.5Mb, or 8.5Mb 32-bit RAM— plus other options!

The CAE/SAR 386 systems run Unix and DOS concurrently, and also run OS/2 and Xenix. Floating point options are available for the Intel 387 chip.

Basic Unix/Xenix systems start at $3,495.

Get a system that fits you perfectly. Call CAE/SAR Systems today for more information.

CAE/SAR Systems, Inc.
P.O. Box 50243
Palo Alto, CA 94303
(415) 949-3816

VERY HIGH RESOLUTION COLOR and MONOCHROME DISPLAY SYSTEMS

- Based on the TMS34010 32 bit graphics CPU - Amazing performance at a reasonable price.
- 800 by 1024 display, 2 bits per pixel for high resolution grey scale.
- PC, XT, and AT Compatible!
- FAST hardware emulation for CGA, Monochrome, and Hercules graphics modes.
- Primary operating software supplied. DGIS based support available for very high performance interfaces to CAD, simulation, and windowing application.

*SPECIAL LIMITED TIME OFFER: PC Tech is offering the complete video system (monitor shown above, graphics adapter card, cables and emulation software) for a special introductory price of $995 plus shipping and handling. Bank card orders welcome.

Designed, Sold and Serviced By:

PCtech

904 N. 6th St.
Lake City, MN 55041
(612) 345-4555

PC, XT, and AT are trademarks of International Business Machines Corp. DGIS is a trademark of Graphic Software Systems, Inc.

ISIS

CP/M MSDOS

COMPLETE SOURCE CODE INCLUDED!

ICXPDS: eXchanger now supports the 5 1/4"iPDS format. Manipulation of ISIS-II files using your computer system was never easier.

ICXMDS: Same as ICXPDS, but for MDS 8" systems.

IMXPDS: Reads/Writes 5" iPDS disks on PC's and AT's.

TELEDPLUS: Enhanced serial file transfer program for CP/M, ISIS, or MS-DOS.

ISE: Emulator gives the CP/M and MS-DOS user access to all the ISIS-II languages and utilities.

ACCELER 8/16: CP/M-80 emulator for MS-DOS. Enables PC's to run ISE. (no source code, V-20 incl.) **$89 each**

$250 any 3 above

UDI: The 8086 ISIS Emulator runs all UDI applications. $300

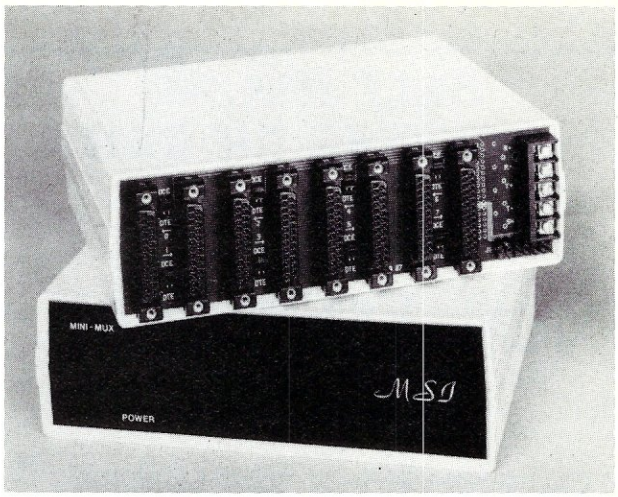
ZAS Development Package: Z-8 and Z-8000 Assembler for CP/M, ISIS, and MS-DOS.

Request a catalog of our products!

Copyrights:CP/M Digital Research, Inc. ISIS-II and iPDS Intel Corp. MSDOS Microsoft

Western Wares 303-327-4898

Box C • Norwood, CO 81423



The Mini-Mux from MSI

have eight channels; ports setable for DCE/DTE configuration; common-channel, four-wire twisted pair; and external power supplies.

The Mini Mux retails for $395 and the Mighty Mux is $495. For more information, contact Method Systems Inc., 3511 Lost Nation Rd., Willoughby, OH 44094; (800) 533-6116, (216) 942-2100.

New Software Products

New FORTRAN Supports 386

Science Applications International Corporation (SAIC) has released SVS FORTRAN-386 for the Compaq Deskpro 386 and compatible computers. SVS FORTRAN-386 allows users to develop large FORTRAN programs (up to 16 MB) and run them in 386 protected mode under control of existing DOS 2.x or 3.x. It also features a native code compiler, a complete runtime I/O system, a source-level debugger, and a VM/RUN 386 runtime monitor. SVS FORTRAN-386 provides full support of Intel 80287/80387 and Weitek 1167 math coprocessors.

SVS FORTRAN-386 is priced at $895, including the source-level debugger and the VM/RUN runtime monitor. For more information, contact SAIC, 5150 El Camino Real, Ste. C-31, Los Altos, CA 94022; (415) 960-5931.

Accurate Quick Basic-to-C Translation

Software Translations, Inc., (STI) has released B-Tran, a Quick Basic-to-C source code translator. B-Tran delivers fast, clean C code from applications written in Microsoft Quick Basic with 11 percent greater speed and more flexibility. Suitable for use with Microsoft, Lattice, and Borland C development systems, the translator is guaranteed 97-percent efficient and can integrate with other C libraries.

B-Tran is priced at $495, and a demonstration disk and manual are available for $25. For more information, contact Software Translations, Inc., 28 Green St., Newburyport, MA 01950; (617) 462-5523. □

Simply the BEST C and Pascal on AT, 386, Sun, Apollo, RT, VAX, 370

"The most rock-solid C compiler in the industry. Superb technical support and portability. Superior code generated."
Gordon Eubanks, Symantec — Q&A (386).

"It simply works, with no trouble, no chasing strange bugs, and excellent warning and error messages … a professional product."
Robert Lerche, Bay Partners.

"For large-scale software development, the highest quality C compiler available on the market today. Pragmas are great. Quality of support is exceptional." **Randy Neilsen, Ansa—Paradox** (DOS,OS/2).

"15% smaller and 15% faster than Lattice C."
Robert Wenig, Autodesk.

"Our software is running anywhere from 30 to 50% faster than when compiled under Lattice." **David Marcus, Micronetics.**

"We switched from Lattice due to a 10% reduction in code size. The compiler is very stable." **Lee Lorenzen, Ventura Software — Ventura Publisher, marketted by Xerox Corp.**

"Best quality emitted code by any compiler I've encountered. Often amazing." **Bill Ferguson, Fox Software — FoxBase (386).**

"Messages sometimes pointed out type mismatches, incorrect-length argument lists, and uninitialized variables that had been undetected for years [in 4.x bsd]." **Larry Breed, IBM ACIS [RT PC].**

"Diagnostics turned up bugs missed by other compilers. Rapid bug fixes by technical support, someone who knew what he was talking about. 80386 code is well optimized."
Tim Addison, Logistics Data Systems.

"386 protected mode support is fantastic, especially the access to large amounts of memory. It's mainframe compute power on a PC." **Dan Eggleston, Viewlogic.**

"The preprocessor supplied with Professional Pascal is quite useful. The code quality and control over segmentation and memory models are superior to MS Pascal." **Bob Wallace, QuickSoft.**

Check Out These Reviews

- **High C ™:**

Computer Language	February 1986, '87	
Dr. Dobb's Journal	August 1986	
PC Magazine	Jan. 27, 1987	(80386 version)
Dr. Dobb's Journal	July 1987	(80386 version)
BYTE Magazine	November 1987	(80386 version)

- **Professional Pascal ™:**

PC Magazine	Dec. 29, 1985	
Computer Language	May 1986	
PC Tech Journal	July 1986	
Journal of Pascal, Ada, & Modula-2	Nov.-Dec. 1986	
BYTE Magazine	Dec. '86, June '87	(80386 version)

Why MetaWare compilers

- They are specifically designed for serious software developers.
- They are reliable and robust: they don't break at every turn.
- Their generated code is the best, or near best, on each architecture.
- Their superior diagnostic messages help you produce better products more quickly.
- Your source can be ported with ease to the most popular systems.
- You can link mixed-language modules from our compilers, others
- You can benefit from high-level, personal technical support.
- You can take advantage of the latest ANSI C extensions, and/or extensive Pascal extensions. **High C** has been tracking the ANSI Standard for two years; **Professional Pascal** will soon have a VS Pascal compatibility switch and several Apollo Pascal ext'ns.
- You can take advantage of the latest 387 and Weitek 1167 support — we have the only compilers with Weitek real mode support.

Power Tools for Power Users

Ashton-Tate: dBase III Plus, MultiMate; Autodesk: AUTOCAD, AUTOSKETCH (8087, '387, Weitek); Boeing Computer Services (Sun); CASE Technology (Sun); CAD/CAM giant Daisy ('86, '386, VAX); Deloitte Haskins & Sells; Digital Research: FlexOS; GE; IBM: 4.3/RT, 4680 OS; Lifetree Software (Pascal): Volkswriter Deluxe, GEM-Write; Lugaru: Epsilon; NYU: Ada-Ed cmplr; Semantec: Q&A; Sky Computers; … (Product names are trademarks of the companies indicated.)

Industrial-Strength

MetaWare C and Pascal compilers are designed for professional software developers. These tools are loaded with options to control them for special purposes. You can adjust the space-time trade-off in code quality. You can adjust external naming conventions to agree with linkers and operating systems. You can specify segment names for segmented architectures, and to help place code or data in particular places for embedded applications. You can select from five memory models for the 8086 family. And on and on.

A Partial List of Optimizations

Common subexpression and dead-code elimination, retention and reuse of register contents, jump-instruction size minimization, tail merging (cross jumping), constant folding, short-circuit evaluation of Boolean expressions, strength reductions, fast procedure calls, automatic mapping of variables to registers (where advantageous), …

"Platform" — Code Quality

Sun,Apollo,SGI	— 18%, 3%, 26% > resident compiler (Dhrystone).
PC: DOS, OS/2	— 3-10% > Microsoft C; 30% > MS Pascal, LatticeC.
386 32-bit DOS	— no competitors, since November, 1986.
286, 386 UNIX	— 66% better than pcc (Dhrystone, 386).
VAX VMS	— ≈ DEC's excellent C and Pascal; better features.
VAX Ultrix	— 19% > pcc (Dhrystone); much > Berkeley Pascal.
RT PC/4.3bsd	— 89% > the original port of pcc (Dhrystone).
370 CMS,UNIX	— much better than any C, and VS Pascal.
AMD 29000	— >40,000 Dhrystones! Available in Q2, cross.

(408) 429-6382, telex 493-0879. ***Since 1979.***

MetaWare™ INCORPORATED
903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060-4429

The Clear Choice for Large Programming Projects — PC Tech J.

© 1987 MetaWare Incorporated. MetaWare, High C, Professional Pascal, and DOS Helper are trademarks of MetaWare Incorporated. Others and their owners are duely respected.

Classifieds

Micro/Systems Journal *accepts Classified Ads. The charge is $6/line (3 lines minimum, 7 lines maximum); 40 characters max./line. Three times frequency $15/line; six times $25/line; non-profit clubs $2/line. Logos, special type, etc. are extra charge. Check must accompany ad copy. Send to M & T Publishing Inc., 501 Galveston Dr., Redwood City, CA 94063.*

GMX20 MICRO: Uniflex OS, C Compiler, VSAM, Editor, et al. 16.7 MHz, 68881 socket, 25MB, 720K floppy. New Cond. $3,500 obo. Marc Talisman: days (714) 582-9100.

HARD TO FIND COMPUTER SUPPLIES For Software Developers and Power Users. IBM style binders & slips. Vinyl B&S. Floppy Pgs., disk envs., custom prtg. MUCH MORE. FREE CATALOG. Anthro Syst., 376 East Saint Charles Rd., Lombard, IL 60148; (312) 629-5160, (800) 332-5669.

"Don't leave home without it."
Credit-card size reference to 56 PC-DOS cmds. $3. Packet Press, 14704-M Seneca Castle Ct., Gaithersburg, MD 20878.

DISK CONVERSION: Most CP/M and MS-DOS formats. 48 hour turnaround. Reasonable rates. Personal service. For information call or write: RH Associates, 2211 Mark Ct., Silver Spring, MD 20910; (301) 587-6230.

MS/J LISTINGS ON DISK
All the computer listings printed in *Micro/Systems* are available on MS-DOS floppy disk. Programs from each issue are available for $14.95 each. For more information, contact:
Tim Trickett
M&T Books
501 Galveston Drive
Redwood City, CA 94063
(415) 366-3600, ext. 221

Kaypro/Osborne CP/M mouse driver, $40, faster and easier cursor movement in WordStar, SuperCalc, graphics, etc. Progressive Products, (408) 265-5490, 1797 Nelson Way, San Jose, CA 95124.

SQL.PAS
Make stack, queue, and list standard data types in Turbo Pascal ($9.95). PSW, Box 10072, McLean, VA 22102-8072.

WE WELCOME YOUR ARTICLES

We are always glad to hear from potential authors who have an interesting tale to tell. If you are interested in contributing an article that relates to local area networks, multiuser systems, or computer programming, please contact us. For example, in upcoming issues we plan to discuss:

- PC Multiuser Operating Systems
- UNIX on the PC
- Graphics on the PC
- 386 Software Development Tools
- Database Options
- Troubleshooting Local Area Networks
- Modem Standards and Compatibility
- High-Capacity Information Storage

We would welcome your contributions on these and related topics. Please contact:

Tom Woolf
Managing Editor
Micro/Systems
501 Galveston Drive
Redwood City, CA 94063
(415) 366-3600

Advertiser Index

Aker Corporation C-3
American Cybernetics 13
Andsor Research, Inc. 15
Austin Code Works 51
Automated Software Concepts Intl. 57
BISS of Louisiana 21
Blaise Computing, Inc. 1
Bytel Corporation 61
C. C. Software 39
CAE/SAR Systems, Inc. 69
Classifieds 72
Communication Horizons 14
Concurrent Controls 5
Controlled Printout Devices 69
Davidge Corporation 50
Digiboard 33
Digital Research Computers 27
Digital Research, Inc. C-2
Ecosoft, Inc. 22
Essential Software C-4
Gimpel 59
Harvard Softworks 47
IGC 9
Interface Group, Inc. 37
Lodden Technology Limited 64
M&T Books 17,67
M-Test Equipment Company 55
Macrotech International 7
Meta Ware Incorporated 50,71
Micro/Systems 56
Micro Way 49
Nanosoft Associates 46
Novell Development Division 2
NWP Intelligent Solutions, Inc. . .48
PC TECH 70
Periscope Co., Inc. 8
Programmer's Shop 35
Qualstar Corporation 39
Quarterdeck Office Systems 65
Raima Corporation 11
Ron Turley Computer Associates .39
Semi-Disk Systems 63
SLR Systems 16
Slicer Computer 69
Software Connections, Inc. 62
Solution Systems 19
Sunny Hill Software 58
Thomas Conrad Corporation 12
Wave Mate, Inc. 40
Western Wares 70
Wyte 21

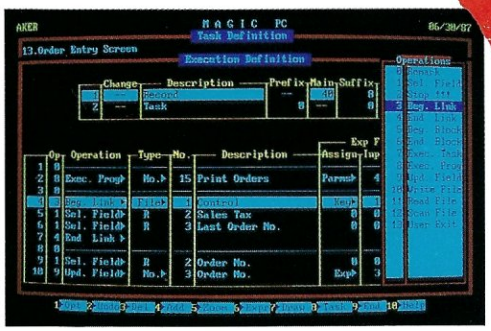
MAGIC PC: A REVOLUTION IN POWER, PRICE & PROGRAMMING SPEED.

You know how database applications are created — by hacking out line after line of time-consuming code. Most DBMS' and 4GL's give you some programming power. But when it comes to serious applications, they keep you bolted to your seat writing mountains of tedious code. And rewriting it all over again with every design change.

Imagine how much faster you'd be if you could replace the painful coding phase with an innovative visual technology which takes only a fraction of the time: Introducing Magic PC—the revolutionary Visual Database Language from Aker Corporation:

High-Speed Programming:

With Magic PC's visual design language you quickly describe your programs in non-procedural Execution Tables. They contain compact programming operations which are executed by Magic PC's runtime engine. You fill-in the tables using a visual interface driven by windows and point-and-shoot menus. One table with 50 operations eliminates writing more than 500 traditional lines of code. Yet with Magic PC you don't sacrifice any power or flexibility.



With a powerful set of high-level non-procedural operations you program at only a fraction of the time.

Maximum Power AND Simplicity:

With Magic PC, you can generate robust DBMS applications including screens, windows, menus, reports, forms, import/export, and much more! Plus, Magic PC has one of the friendliest user interfaces you've ever seen. Using Magic PC you can look-up and transfer data through a powerful Zoom Window system. Magic PC even lets you perform command-free queries.

Btrieve Performance:

Magic PC incorporates Btrieve, the high-performance file manager from SoftCraft. This gives you exceptional access speed, extended data dictionary capabilities, and automatic file recovery!

Virtually Maintenance-Free:

With Magic PC you can modify your application design "on the fly" without any manual maintenance. Magic PC automatically updates your programs and data files on-line! This also makes Magic PC an ideal tool for prototyping complete applications in hours instead of days.

FREE Networking:

Magic PC comes complete with LAN features. Develop multi-user applications for your LAN with Magic's file and record-locking security levels.

Stand-Alone Runtime:

Distribute your applications and protect your design with Magic PC's low cost runtime engine.

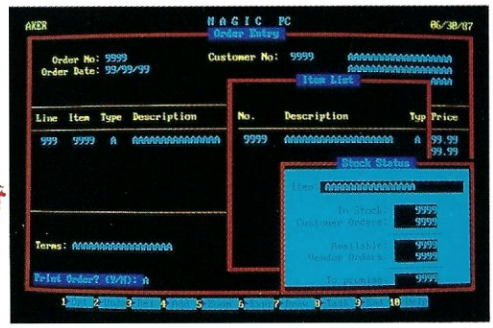
All For Only $199:

Best of all, Magic PC is an unbeatable bargain. For a limited time, Magic PC's price has been reduced to only $199! Yes, this is the same Magic PC that normally lists for $695! And Magic PC *eliminates the need for a separate DBMS, compiler, or application generator. It comes complete with all the tools you need* to develop your own database applications instantly.

$199 – With A Money-Back Guarantee!

For a limited time, you can get Magic PC for only $199. And even at this low price, Magic PC is risk-free. If you're not completely satisfied, simply return it within 30 days and we'll buy it back (less $19.95 restocking fee). And if you'd like a preview, Magic PC's Tutorial Demo is available for just $19.95.

But you'd better hurry – Magic PC's special $199 price won't last long!



Pop-up Zoom Windows run multiple programs per screen — with point-and-shoot data transfer between windows!

Join The Magic PC Revolution

To unleash your DBMS design power, order your $199 copy of Magic PC right now by calling toll-free or returning the coupon below.

ORDER NOW: CALL (800) 345-MAGIC

In CA (714) 250-1718

"Magic PC's data base engine delivers powerful applications in a fraction of the time. . . there is truly no competitive product."

Victor Wright – PC Tech Journal

Also recommended by: *PC Magazine, PC World, PC Week, Computer Language, Data Base Advisor,* and many other publications worldwide.

MAGIC PC
The Visual Database Language
by AKER

Yes! I want to generate powerful applications much faster!

☐ Rush me my copy of Magic PC at the special promotional price of $199 (add $10 P&H, and tax in CA. International orders add $30). I understand I can return Magic PC for a refund within 30 days, if I'm not completely satisfied.*

☐ Rush me a copy of Magic PC Tutorial Demo at $19.95 (add $5 P&H, and tax in CA. International orders add $15).

Name _____

Company _____

Street Address (no POB) _____

City _____ State _____ Zip _____

☐ Check enclosed Charge to my: ☐ VISA ☐ ☐ AMERICAN EXPRESS

Account No. _____

Acct. Name _____ Exp. Date ____/____

Signature _____

Return to: **Aker Corp.,** *18007 Skypark Cir B2, Irvine, CA 92714*

MS

System requirements: IBM PC, XT, AT, PS/2 or 100% compatible with 512K RAM, hard disk and DOS 2.0 or later. 5¼" format, not copy protected. Dealer pricing available. *Return policy valid in US only.

Aker, Magic PC, The Visual Database Language are trademarks of Aker Corporation. All other trademarks acknowledged. © Copyright 1987, Aker Corp.

How A C Programmer Became A Screen Star

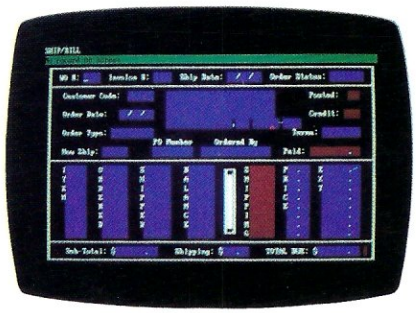
Screens, the Visible Part of Your Program.

A program is often judged by how well the screens are executed. However, the real creativity lies in what goes on behind the screens.

ScreenStar is a product that allows your real creativity to light up the screen. It reduces costly screen, window, and data validation development time.

You Take the Bows, We Write the Code.

Our natural drawing commands allow you to paint any screen imaginable. Press one key when you are satisfied and ScreenStar produces concise, commented, ready-to-compile code. This allows immediate testing of the I/O screens, including smooth, even scrolling between multiple screens.



Create or capture complex screens with data-entry filters built in.

If all ScreenStar did was turn screens into code it would be a useful tool. Yet ScreenStar also permits a wide range of field types. Some of the choices include date, alphanumeric, telephone, yes/no, dollar, time and user-definable fields.

Other valuable data-entry filters are built in, such as required field, display only, and many others. All screen fields are generated with error-checking routines.

ScreenStar Not Only Captures Your Imagination, It Captures Screens.

The memory-resident capture program converts any screen into a ScreenStar file in seconds, including those generated by programs like Dan Bricklin's Demo Program.

ScreenStar Sets the Stage for Windows.

ScreenStar comes with a complete window generating library. You design the help screens and pop-up windows. Essential ScreenStar windowing functions tie them together in one smooth package.

Curtain Call.

They may not ask for your autograph, but they will want to know how you did those screens. Screenstar is more than a screen-painting program. It is a screen processor. No professional programming environment will be complete without this product.

We know you will enjoy using ScreenStar. However, should you give it less than rave reviews, return it within 30 days for a full refund.

★ Interactive screen painting and subsequent code generation.

★ Multiple screen design and scrolling.

★ TSR screen capture program, works with any program including Dan Bricklin's Demo Program.

★ Complete window design including overlapping window functions.

★ Screens are compressed into data structures, and remain a permanent part of the program. No messy data files to look for.

Price - $99

W/Source add $99

Audition Our Product Today. Call:

(201) 762-6965

Essential Software
South Orange Plaza
76 South Orange Ave., Suite 3
South Orange, NJ 07079

ScreenStar is a trademark of Essential Software Inc.
Dan Bricklin's Demo Program is a trademark of Software Garden Inc.